4-2011

COMPUTATIONAL COMPLEXITY OF APPROXIMATE AND PRECISE DATA WITH CONSTRAINT AUTOMATON

Dipty Singh
University of Nebraska-Lincoln, diptysingh@gmail.com

COMPUTATIONAL COMPLEXITY OF APPROXIMATE AND PRECISE DATA WITH CONSTRAINT AUTOMATON

by

Dipty Singh

A THESIS

Presented to the Faculty of

The Graduate College at the University of Nebraska

In Partial Fulfillment of Requirements

For the Degree of Master of Science

Major: Computer Science

Under the Supervision of Professors Peter Z. Revesz

And

Jean-Jack M. Riethoven

Lincoln, Nebraska

April, 2011

COMPUTATIONAL COMPLEXITY OF APPROXIMATE AND PRECISE DATA WITH CONSTRAINT AUTOMATON

Dipty Singh, M.S

University of Nebraska, 2011

Advisors: Peter Z. Revesz and Jean-Jack M. Riethoven

The DNA molecules packaged in structures called chromosomes within the cells of living organisms encode hereditary information that is passed on to their offspring. Using transcription and translation, the genes within these DNA molecules help in protein synthesis. Thus chromosomal DNA serves as a blueprint for the chemical processes of life.

In order to analyze a DNA sequence by currently available technology, we have to cut it into small fragments, e.g. by using restriction enzymes. The application of different restriction enzymes to the multiple copies of the same DNA sequence generates many overlapping fragments. In order to construct the original DNA, these fragments need to be sequenced and assembled. This problem of finding the original order of the fragments is called the genome map assembly problem.

This research proposes a constraint automaton solution to solve the genome map assembly problem for both error prone and error free data. Plasmid vectors puc57, pKLAC1-malE, pTXB1 and phage vector Adenovirus2, having a size in base pairs of 2710, 6706, 10153 and 35937 respectively, were used to prove that computational time for solving genome map assembly problem using constraint automaton solution is linear with both precise and approximate data.

Dedicated to my parents, Sharada Singh and Gorakh Bahadur Singh, and my husband Kiran Bikram Khadka

Table of Contents

List of Figures

List of Tables

Chapter 1

1. Introduction

The DNA molecules packaged in structures called chromosomes within the cells of living organisms encodes hereditary information that is passed on to their offspring. Using transcription and translation, the genes within these DNA molecules helps in the production of protein. Thus chromosomal DNA serves as a blueprint for the chemical processes of life.

DNA sequences range from thousands to billions of base pairs. However, currently available sequencing machines can only handle a couple of thousand base pairs at a time. So the long DNA sequences have to be cut into smaller subsequences using restriction enzymes. After cutting the DNA, small fragments just float randomly in the solution, losing all the information about the original order of the sequence. After the subsequences are sequenced and analyzed, they have to be arranged and assembled to obtain the original sequence. This process is called *Genome Map Assembly* (Harley and Bonner) and the problem of executing the Genome Map Assembly process is called the *Genome Map Assembly Problem (GMAP)* [25]

This thesis improves on the existing *constraint automata solution* [24] to solve the Genome Map Assembly Problem for both precise and error-prone data.

1.1 Constraint Automata

Based on conditions that are described using constraints on variables, *constraint automata*, are used to control the operation of a system. It has to find the set of *reachable configurations*, which is the set of states and state values that the constraint automaton can enter. This is one of the important problems in constraint automata.

A constraint automaton can be simplified using *reduction rules*, which allow us to either rewrite complex constraints into simpler ones or eliminate some transitions from the constraint automaton.

It consists of set of states, a set of state variables, transitions between states, an initial state, and the domain and initial values of the state variable. Each transition consists of *guard* constraints (set of constraints) followed by assignment statements. In constraint automata, the assignment statements are shown using the symbol '=' and guards are followed by questions marks, e.g. a>=100?

If there is a transition whose guard constraints are satisfied by the current values of the state variables, a constraint automaton can move from one state to another. In addition to the state variables, the transitions of a constraint automaton may contain variables. Some of the values for these variables must be found such that the guard constrains are satisfied and the transition can be applied. For this reason, these variables are said to be *existentially quantified* variables.

By sensing the current value of a variable, a constraint automaton can interact with its environment. This is expressed by *read(x)* command on a transition between states, where x is any variable. This command can appear either before or after the guard constraints, which updates the value of x.

1.1.1 Illustration of Constraint Automata

This example is from. [26] A ferry takes passengers across a river between shores *Shore-A* and *Shore-B* and stops occasionally at an island *Island-C*, which is visited by few passengers.

A ferry visits the island according to the following rules:

- If there are any passengers waiting to stop at *Island-C*, the ferry then visits the island on the third crossing since the last visit.
- If there are at least ten passengers who are waiting to stop at *Island-C*, then the ferry may visit the island earlier, but never on two subsequent crossings.

Assuming we know how many new passengers who want to go to *Island-C* will board the ferry each hour, we have to find all possible combinations of location and number of passengers waiting in the ferry for one of two crossings.

The constraint automaton is shown in Figure 1.1. There are three states for the three possible locations of the ferry; *Shore-A, Shore-B and Island-C*. Following are the state variables:

- n – the number of crossings since the last visit to *Island-C*
- x_0 *and* x_1 – the number of passengers waiting to visit C since t= 0 and t = 1 respectively
- x – the number of passengers freshly boarding with tickets to the island

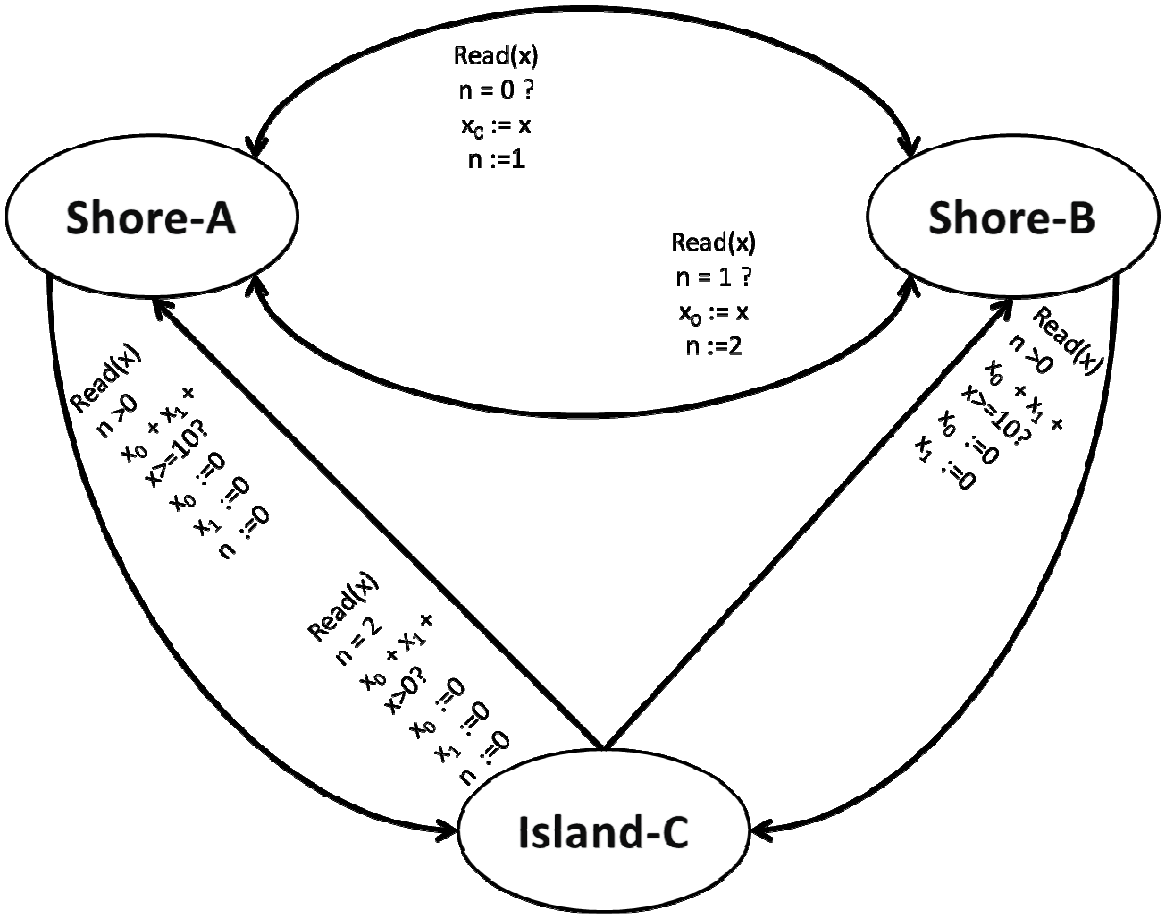

Figure 1-1: The Ferry System [26]

1.2 Genome Map Assembly Problem

Currently available sequencing machines can only handle DNA fragments of a couple of thousand base pairs. So the long DNA sequences have to be cut into small fragments using restriction enzymes, which results in the loss of information about the original order of sequence. After these fragments are sequenced and analyzed, they have to be arranged and assembled to obtain the original sequence. This process is called *Genome Map Assembly* (Harley and Bonner). The problem of reassembling these fragments from

data that is incomplete, imprecise, ambiguous and often contradictory is known as the *Genome Map Assembly Problem (GMAP*) [25]

1.3 Biological Background

The main actors in the chemistry of life of both complex and simple organisms are molecules called proteins and nucleic acids. “Proteins are responsible for what a living being is and does in a physical sense. Nucleic acids encode the information necessary to produce proteins and are responsible for passing along the recipe to subsequent generations.” [29]

A protein is the chain of molecules called *amino acids*. Twenty different amino acids are commonly found in proteins. Each protein has a unique amino acid sequence which determines its specific shape and functions.

1.3.1 Nucleic Acids

There are two kinds of nucleic acids in a living organism; *ribonucleic acid (RNA)* and *deoxyribonucleic acid (DNA).*

DNA is a hereditary material in most of the living organisms. Information in DNA is stored in the form of code which is made up of four chemical bases known as: adenine (A), guanine (G), cytosine (C), and thymine (T). Information required for building the organism is determined by the sequence of pairing between these bases. DNA base pairs are formed by combining A with T, and C with G. DNA’s strand orientation is by convention 5’ to 3’ unless otherwise specified. Base pairs abbreviated as *bp* provide the unit of length of a DNA molecule.

RNA molecules are like DNA molecules but uracil (U) is present instead of thymine (T) and unlike DNA, RNA doesn't form a double helix.

1.3.2 Genes and the Genetic Code

A gene is the segment of DNA found in a chromosome which is the basic physical and functional unit of heredity. As mentioned earlier, a protein consists of a chain of amino acids. So to specify a protein we have to specify the amino acids it contains and this is done by the DNA in the gene. It uses a triplet of nucleotides known as *codon* for each amino acid.

1.3.3 Protein Synthesis

A copy of the gene is made on a RNA molecule, resulting in *messenger RNA* (mRNA). This RNA will have the complement of the strand of DNA that it has been transcribed from, (which is same as the opposite strand) with *T* replaced by *U*. This process of converting DNA to mRNA is called *transcription.* The mRNA is then decoded by the ribosome to produce a specific amino acid chain. This process of implementing the genetic code in a protein is called *translation*. Genes are composed of alternating parts of *intron*s and *exons.* Introns do not take part in protein synthesis and are removed from mRNA after transcription.

1.3.4 Chromosome

A chromosome is a thread-like structure of DNA found in the nucleus of a cell. It is made up of a single DNA which is tightly coiled several times around proteins known as

histones. Chromosomes are visible under a microscope only during the cell division process. Therefore most of the research on chromosomes is performed during the cell division. A chromosome is divided into two parts or arms by a constriction point known as the centromere. The shorter arm is called p arm whereas the longer arm is called q arm. It is the location of the centromere that determines the shape of a chromosome.

1.3.5 Genome

"A genome is all of a living thing's genetic material. It is the entire set of hereditary instructions for building, running, and maintaining an organism, and passing life on to the next generation" [10] It is divided into chromosomes, which contain various types of functional elements of which genes are the most widely known.

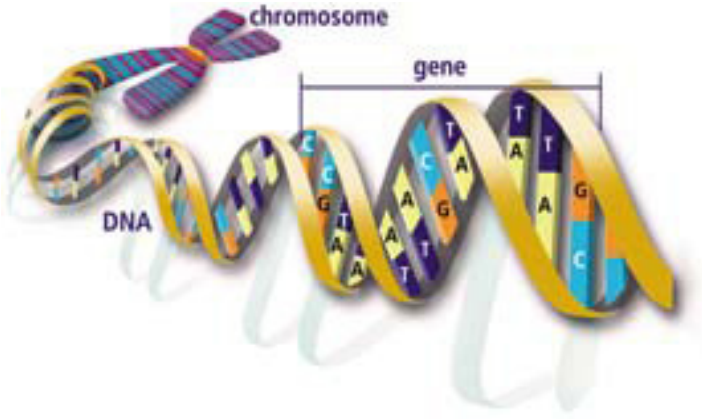

Figure 1-2: Genome [35]

1.3.6 Genome Sequencing

Genome sequencing is the process of finding the precise order of DNA nucleotides in a genome; the exact order of adenine, cytosine, guanine, and thymine that make up an organism's DNA.

Genome sequencing allows scientists to sequence genes and genomes. Due to the limitation of how many bases can be sequenced in one experiment, DNA has to be broken into smaller fragments before they can be sequenced and reassembled.

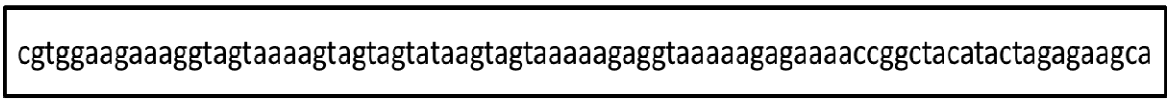

Figure 1-3: Genome Sequencing

1.3.7 Genome Mapping

Genome mapping is the process of finding the approximate position of genes in a genome without getting into the details of the actual sequence. It is a graphical representation that helps to find out where you are and how to get to where you want to go. It identifies the order of the specified subsequences in the genome. The genome map contains various landmarks identified with a series of letters and numbers, which help the researchers, find where specific pieces of DNA belong in the overall genomic jigsaw puzzle.

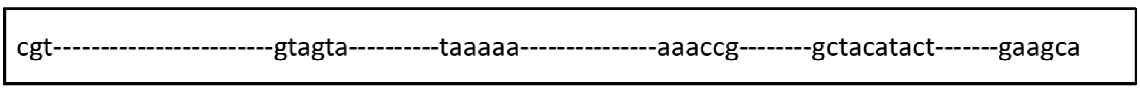

Figure 1-4: Genome Mapping

1.3.8 Restriction Endonucleases

Restriction endonucleases, commonly called *restriction enzymes,* are nucleases that are made by bacteria to protect themselves from a virus. [3] All kinds of bacteria make restriction enzymes and these enzymes are part of their immune system. These enzymes can digest the DNA of a virus that attempts to inject its DNA and reproduce into the bacterial cell. This action results in the death of the invading virus. Molecular biologists have purified these enzymes for analyzing chromosome structure, sequencing very long molecules, isolating genes, and creating new DNA molecules that can be cloned. These enzymes recognize specific nucleotide sequences and cleave DNA at a specified location. All restriction enzymes are named after the bacterium from where they are isolated. The cutting sites (restriction sites) have the same name as the relevant enzyme. The following are some common restriction enzymes.

Restriction Enzymes	**Restriction Sites**
Cla1	AT^CGAT
EcoRI	G^AATTC
HindIII	A^AGCTT
PstI	CTGA^G

Table 1-1: Common Restriction Enzymes

(^ is the restriction site where the sequence is cleaved)

1.3.9 Fingerprints

It is hard to compare and match long nucleotide sequences. A short-cut method what is often much faster is *Fingerprinting. Restriction site analysis* and *hybridization* are the two most popular method of getting fingerprints. In restriction site analysis, one or more

restriction enzymes are applied to the DNA sequence and the lengths of the resulting fragments are measured, these lengths thus serve as the "fingerprint" of each subsequence. In hybridization fingerprinting, certain small sequences are checked to see if it binds to fragments, the subset of such small sequences that binds to the fragment serves as its fingerprint.

1.4 Related Work

1.4.1 Overlap Multigraph

The *overlap Multigraph* [30] is one of the ways of doing fragment assembly. This method is usually illustrated by a graph where each node is one of the subsequence. A directed edge from node *A* to the different node *B* with weight $k >= 0$ exists if the suffix of *A* with *k* nucleotide is a prefix of *B*.

$$suffix(A,k) = prefix(B,k)$$

Here weight k represents overlaps between fragments and overlaps are used to join fragments into longer strings, sharing the overlapping parts. For the edge to exist we must have $|A| >= k$ *and* $|B| >= k$.

The joining of two fragments based on overlaps means they come from the same region in the DNA. The size of the overlaps determines whether it is a strong enough evidence to join two fragments; the bigger the size of overlap, the stronger the evidence to join. We can also impose a minimum acceptable amount of overlap, and ignore all the edges with the weight less than this value.

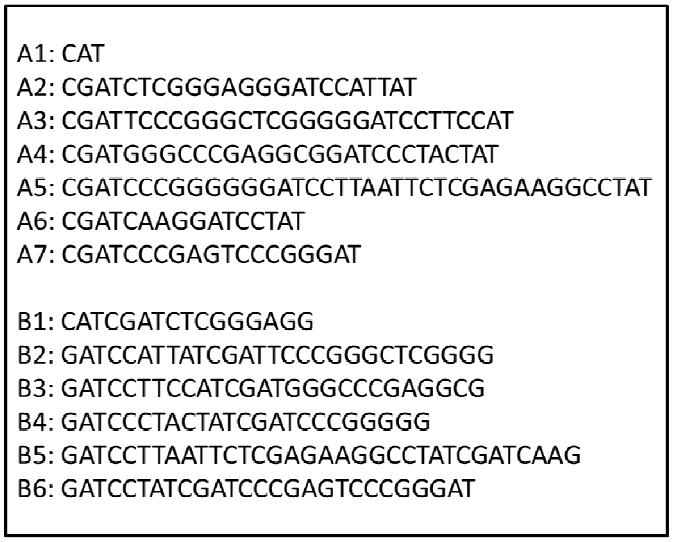

Figure 1-5: Fragments of A and B [24]

In this example, *F* is {*A1,A2,A3,A4,A5,A6,A7,B1,B2,B3,B4,B5,B6*}. Only the edges with positive weight are drawn. As the name implies, there are many edges possible between two nodes. As seen in the Figure 1-6, there are two directed edges from B3 to A4.

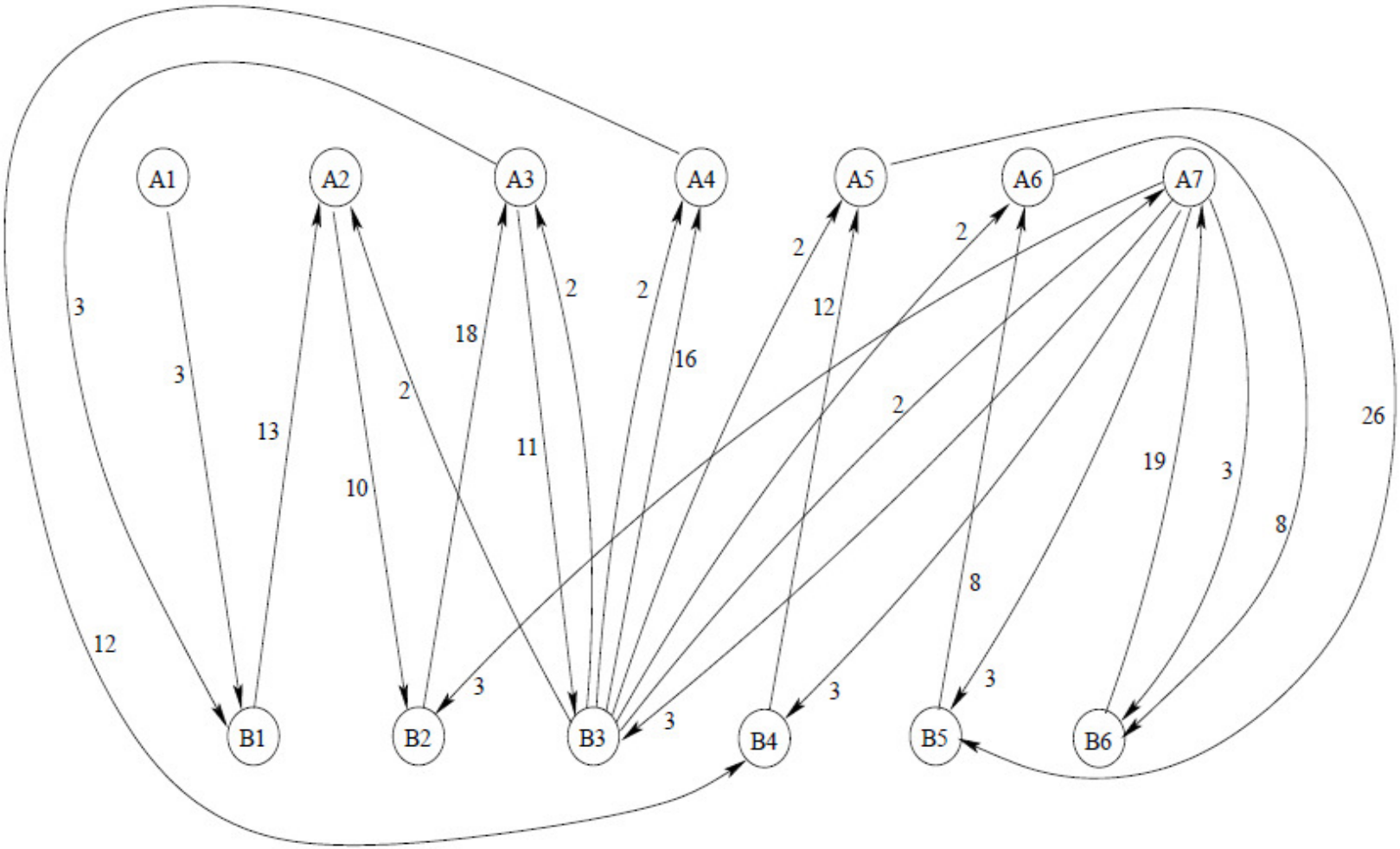

Figure 1-6: Overlap Multigraph for F [24]

"Hamiltonian path, also called a Hamilton path, is a path between two vertices of a graph that visits each vertex exactly once" [37] Hamiltonian path in the overlap multigraph is a *shortest common superstring* of the sequence *F* and hence the solution of genome sequencing problem. Here our goal is to maximize the weight; we simplify the multigraph and consider only the heaviest edge between every pair of nodes, discarding the others.

The following Hamiltonian path can be derived from the above overlap multigraph.

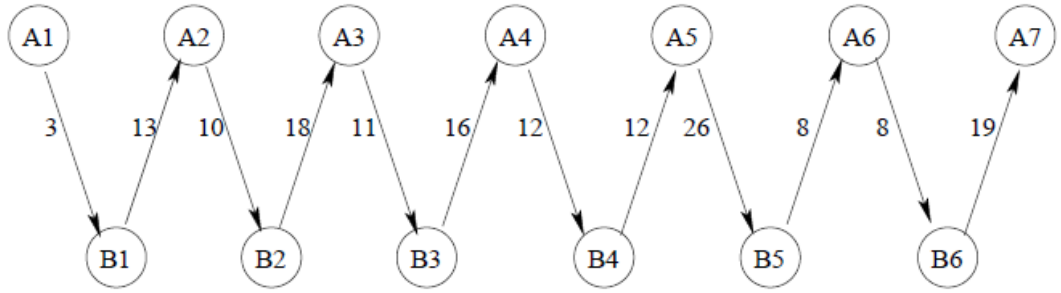

Figure 1-7: Hamiltonian path in the Overlap Multigraph [24]

The above Hamiltonian path is an ideal case. A search for a graph by greedy algorithm would always choose the edges with the highest label. From figure 1-7 we can see that, greedy algorithm will pick *A5* because it is the node with the highest label, 26. Greedy search will start at *A5,* then *B5* and so forth. But *A5* is the failure node. Figure 5 shows the failure assemble genome using greedy algorithm in overlap multigraph.

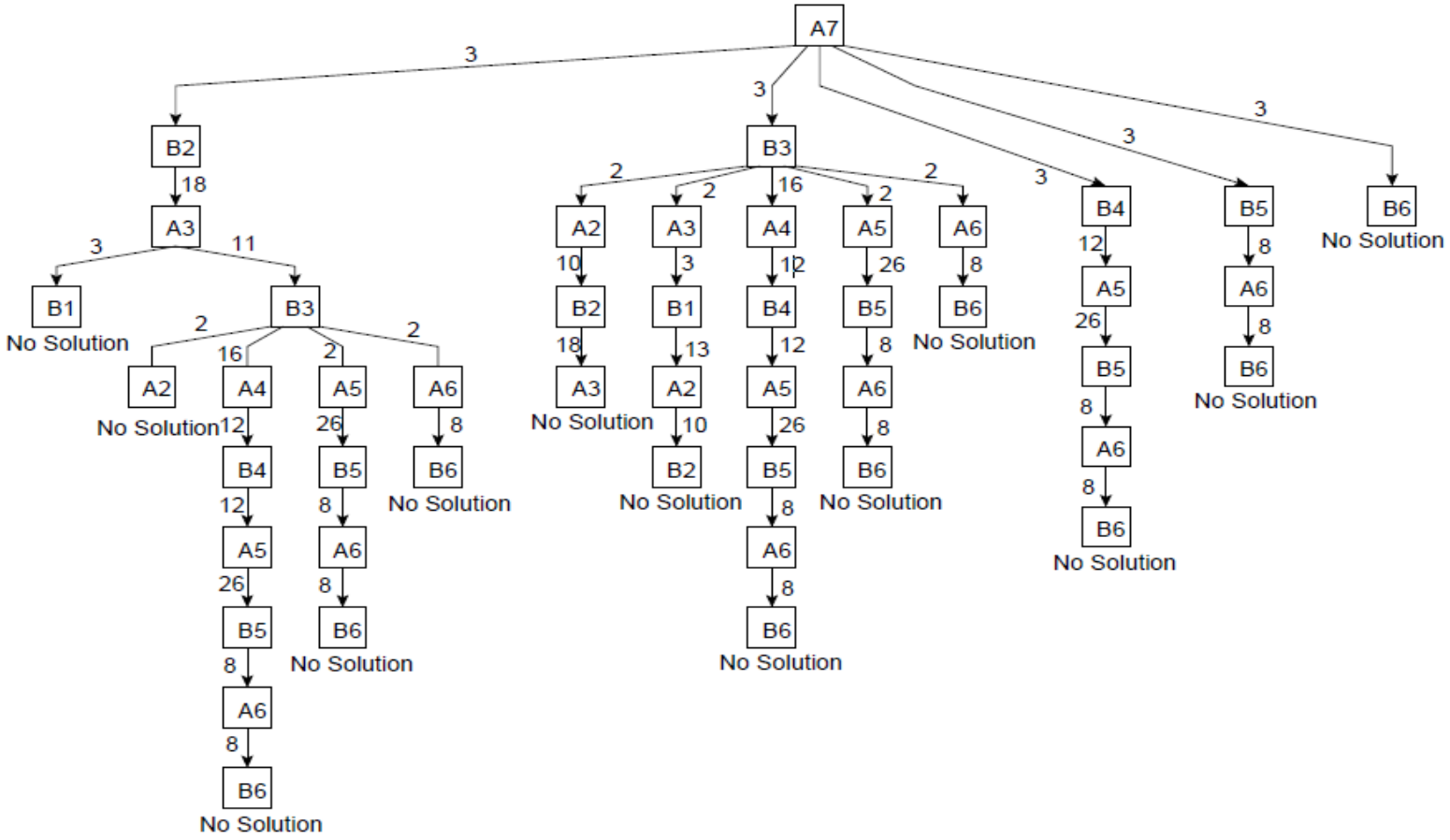

Figure 1-8: Failure with Greedy search from node A7 of the overlap multigraph [24]

From the above example it is clear that greedy algorithm will not always return the *shortest common superstring*. Using the *shortest common superstring* through the Hamiltonian path problem is NP – complete [30]

1.4.2 Other alternatives

Several approaches have been implemented in an effort to solve the genome map assembly problem, and most of them rely on shotgun sequencing. In this sequencing method, DNA is first shredded into smaller fragments and these are then reassembled into their original order based on overlaps, ultimately yielding the complete sequence. DNA is either shredded using restriction enzymes or mechanically by shearing. And finally alignment of the sequences of overlapping fragments is done by computer. [5]

There are many variations of the overlap multigraph method depending on how the subsequences are generated and how the fingerprints of these subsequences are recognized. Mainly there are two approaches to generate fingerprints of the subsequences; the restriction fragment fingerprint and hybridization fingerprinting. In the restriction fragment fingerprint, the subsequences are obtained using a different combination of restriction enzymes and the length is measured using gel electrophoresis, which serves as a fingerprint. In hybridization fingerprinting, subsequences are exposed to a number of probes, and it is determined which of these probes hybridizes to the clone.

Coulson, A, et al. [6], Olson, MV, et al [22], Siegel, A F, et al [31], Wong,G K, et al [38], Revesz [27], are all based on restriction fragment fingerprinting where fragments

were assembled based on overlaps of randomly chosen subfragments. Using this fingerprinting method, some fragments may be lost and also the gel electrophoresis only gives approximate length. So instead of finding the perfect solution, fragments that best fit the noisy data are picked.

Green and Green[13], Harley and Bonner[14] and Lander and Waterman[17] work were based on hybridization fingerprinting. The hybridization fingerprint is subject to *false positives*, in which a probe is recorded incorrectly while hybridizing to a subsequence and *false negatives* in which the opposite error is made. [16]

In both of these approaches, random substrings are picked from the DNA. This procedure did not produce a complete physical map. Since most of the regions were not covered by any randomly picked substrings, it produced gaps instead.

Karp [16] shows several variations of these overlap multigraph problems to be NP-complete. Revesz's *Big-Bag-Matching* abstraction [26] for GMAP is also NP-complete.

Chapter 2

2. The Constraint Automata for GMAP

The algorithm of the constraint automata for GMAP can be described easily, if we define GMAP as a *Big-Bag Matching Problem* [25][28]. The Constraint Automata Solution proposed by Ramanathan and Revesz [24] has been modified to deal with both error prone and error free sub fragments of DNA.

2.1 Big-Bag-Matching Problem

"A *bag* is a multiset, a generalization of a set in which each element can occur multiple times. A big-bag is a multiset whose elements are bags that can occur multiple times. [25]

Big-Bag-A	Big-Bag-B
A = {A1, A2, A3}	B = {B1, B2, B3}
A1 = {23, 12, 23}	B1 = {22, 19, 18}
A2 = {15, 16, 18}	B2 = {19, 12, 23}
A3 = {21, 18, 17}	B3 = {61, 12, 9}

Figure 2-1: A1, A2, A3, B1, B2, B3 are bags within A and B big-bags

Each permutation of the bags and permutation of the elements of each bag within a big-bag is called a *presentation* [24] [25] There can be several different presentations of a single big-bag. "The big-bag matching decision problem (BBMD) is the problem of deciding whether two big-bags match. The big-bag matching problem (BBM) is the

problem of finding matching presentations for two given big-bags if they match" [24][25]. We use the concept of BBM to solve the GMAP.

2.2 Fingerprint data

A modified version of Revesz's Fingerprint [25] data has been used to collect fingerprints of input data. Instead of using three restriction enzymes only two enzymes have been used. The fingerprints collected will later be used as an input for Constraint Automata Solution for GMAP. The methodology was as follows:

1. An original DNA sequence was copied.
2. Restriction enzyme "*a*" was applied to the copied sequence which created several fragments of varying lengths depending on the restriction site.
3. Individual fragments were separated.
4. Restriction enzyme "*b*" was applied to the separated fragments from step 3, producing sub-fragments.
5. The length of individual sub-fragments was measured.
6. Steps 1-5 were repeated, but restriction enzyme "*b*" was applied first and restriction enzyme "*a*" was applied consequently.

All the elements from first copy of DNA are stored in *Big-bag-A* and elements from the second copy of DNA are stored in *Big-bag-B*.

2.3 Constraint Automaton for GMAP

The constraint automaton purpose in this thesis is the modified version of the initial two versions purposed by Revesz [25][27] and Ramanathan [24]

2.3.1 Constraint Automaton solution for GMAP without Backtracking

The automaton has the following states:

- INIT – this is where the automaton begins
- A-ahead – if A bag is ahead
- B-ahead – if B bag is ahead
- HALT – if the solution is found

The automaton has the following state variables:

- *UA*: set of A bags which has not been used yet
- *UB*: set of B bags which has not been used yet
- S: set of elements by which either A or the B bag is currently ahead

The automaton starts in the INIT state and tries to reach the HALT state. The automaton moves from left to right by adding either an A bag or a B bag. If *A bag* is greater than B bag, it goes to A-ahead state. Else, it goes to B-ahead state. If they are equal, it goes to INIT state and starts the automaton with remaining *UA* and *UB*. When *UA*, *UB*, and S are empty and all the bags are used, automaton goes into HALT state and stops.

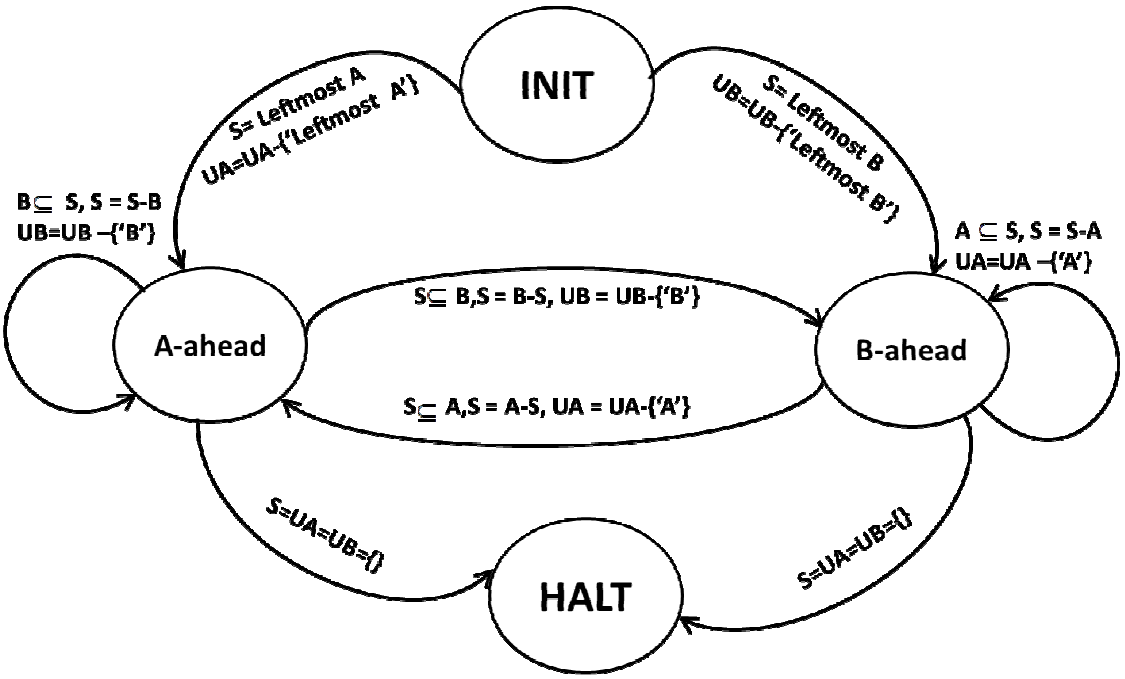

Figure 2-2: Constraint Automaton for GMAP -Version 1

This algorithm does not use backtracking. If it does not find the elements in S in both A-bag and B-bag it goes back to INIT state and starts the automaton all over again. This makes it very inefficient.

2.3.2 Constraint Automaton solution for GMAP with Backtracking

This is the second version of constraint automaton solution for GMAP purposed by Revesz and Ramanathan [24]. The solution presented here has many improvements when compared to the previous version. One new state and a few state variables were added to make the automaton more efficient.

The following are the states used in this version of constraint automaton.

- INIT – this is where the automaton begins

- A-ahead – if A bag is ahead
- B-ahead – if B bag is ahead
- Backtrack – if solution is not yet found but the *S* is empty
- HALT – if the solution is found

The automaton has the following state variables:

- *UA*: set of A bags which has not been used yet
- *UB*: set of B bags which has not been used yet
- *CurrBag* : current bag, which is the *SelBag* from previous state
- S: set of elements by which either the A or the B bag is currently ahead
- *Options*: set of bags from which the next bag can be chosen
- *SelBag*: bag that as selected from the *Options* as the next bag
- *Choices*: set of remaining bags that were not picked from *Options* besides *SelBag*
- *Cflag*: 0 if the *Choices* is empty and 1 if the *Choices* is not empty

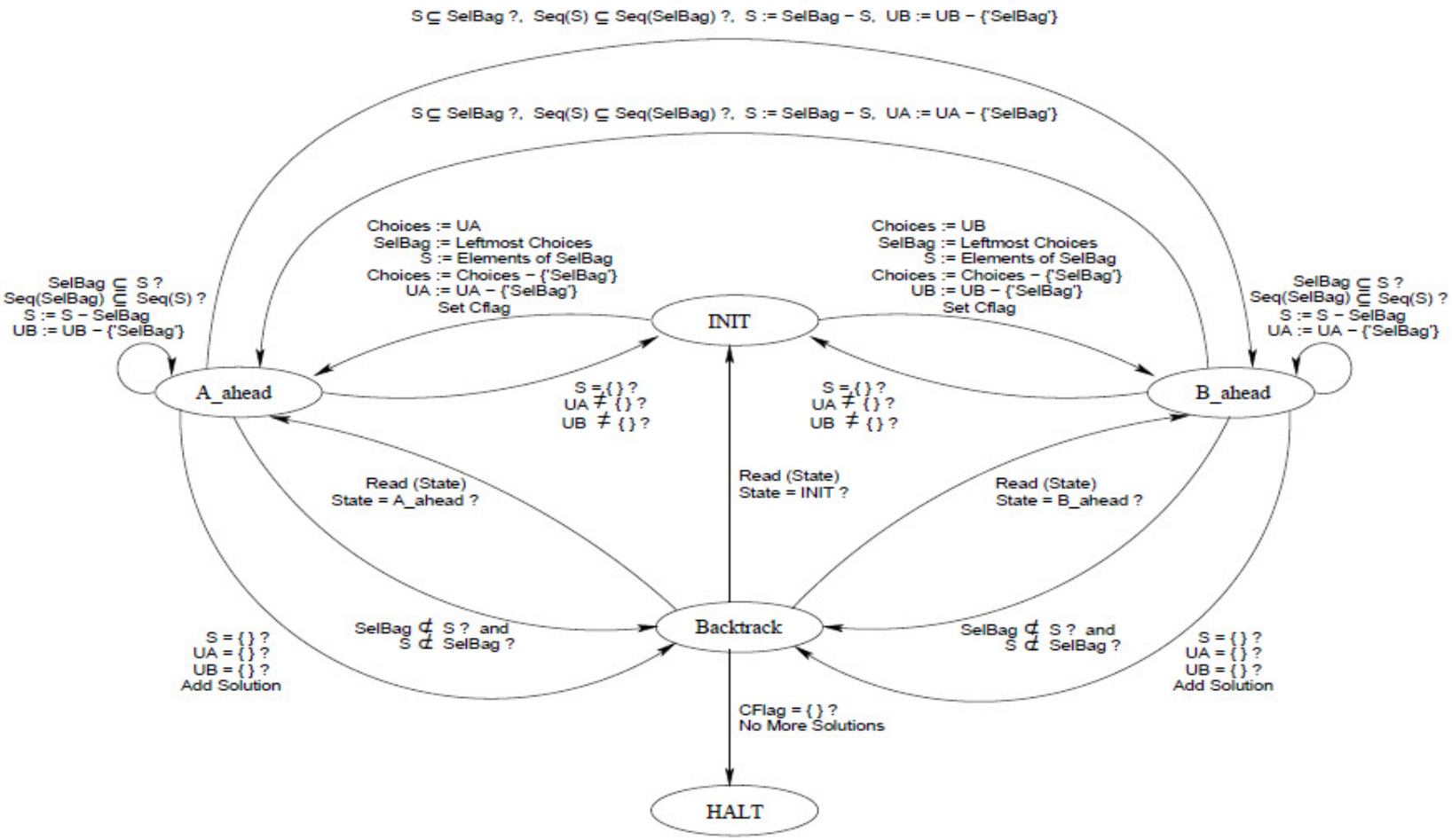

Figure 2-3: Constraint Automata Solution for GMAP Version 2[24]

At first the automaton is in the INIT state. The next bag is chosen from the big-bag A or B which has the least number of bags. If the next bag is picked from the A list, then, *Options* will contain elements of *UA*. But if next bag is picked from the B list, then *Options* will contain elements of *UB*. The leftmost bag from *Options* is removed and set to *SelBag* and elements of it are set to S. The remaining bags of *Options* are copied to *Choices*. *Cflag* is set to 0 if *Choices* is empty, if *Choices* is not empty it is set to 1. The next bag is picked from the bags which are either a subset or superset of S.

The automaton goes to the A-ahead or B-ahead state whichever has the least number of elements so far. It can also remain on the same state if the other big-bag still has a greater number of elements. If the automaton does not find a bag which is either a subset or superset of S, it goes to Backtrack state.

If solution is found, UA, *UB* and S will be empty and it goes to HALT state.

This version of constrain automaton solution for GMAP is more efficient than the earlier version and the time complexity for solving GMAP is linear [24]

But while fingerprinting the fragments of DNA sequence, there are chances of mismeasuring the length of each sub-fragment and getting the wrong fingerprint. This automaton works well if data doesn't have any error. It does not handle data with errors.

2.3.3 Constraint Automaton for Error Prone Data

The earlier version of the constraint automaton did not handle data with errors the purpose of this thesis is to propose the constraint automaton for solving GMAP for both precise and approximate data. The proposed constraint automaton handles data with errors up to a given threshold value.

The new constraint automaton adds new state and state variables to the existing constraint automaton. Following are the states:

- Error-Check – check to see if input data has any errors
- Replace-Error- – if there is an error, replace it with the mean of two mismatched data elements from both A and B bags
- INIT – this is where the automaton begins
- A-ahead – if A bag is ahead
- B-ahead – if B bag is ahead
- Backtrack – if solution is not yet found but the *S* is empty
- HALT – if the solution is found or if the error is greater than error tolerance value

The automaton has the following state variables:

- Error: difference between two mismatched values from the A and B bags
- ErrorTolerance: the specified error tolerance
- *Length-A* : all the elements of S that belongs to *Big-Bag-A*

- *Length-B* : all of the elements of S that belong to *Big-Bag-B*
- *UA*: set of A bags which has not been used yet
- *UB*: set of B bags which has not been used yet
- *CurrBag* : current bag, which is the *SelBag* from previous state
- S: set of elements by which either the A or the B bag is currently ahead
- *Options*: the set of bags from which the next bag can be chosen
- *SelBag*: the bag that as selected from the Option as the next bag
- *Choices*: set of remaining bags that were not picked from *Options* besides the *SelBag*
- *Cflag*: 0 if *Choices* is empty and 1 if *Choices* is not empty

The automaton goes through the following steps to solve the GMAP.

1. The input to the constraint automaton is the fingerprints of DNA fragments. Bags of Big-Bag A and Big-Bag B are fed to the constraint automaton. The automaton starts with Error-Check state where input data are checked for any errors. The error occur when the length of sub-fragments of DNA is mismeasured, which results in two different sets of elements in big bag *A* and *B*. To check for any errors, each element of *Bags* within *Big-Bag-A* is compared with the elements of *Bags* within *Big-Bag-B*. If the elements do not match the input data has some error, otherwise the input data is error free.
    a. If there is no error in the data it goes to INIT state.
    b. If the error is more than specified error tolerance value, then it goes to HALT state.

c. If the error is less than or equal to the specified error tolerance value, it goes to Replace-Error state.

2. If the data has some error, it goes to Replace-Error state. The idea is to replace the wrong data with the mean of two mismatched data, so that it will have the same set of data in both *Big-Bag-A* and *Big-Bag-B*.

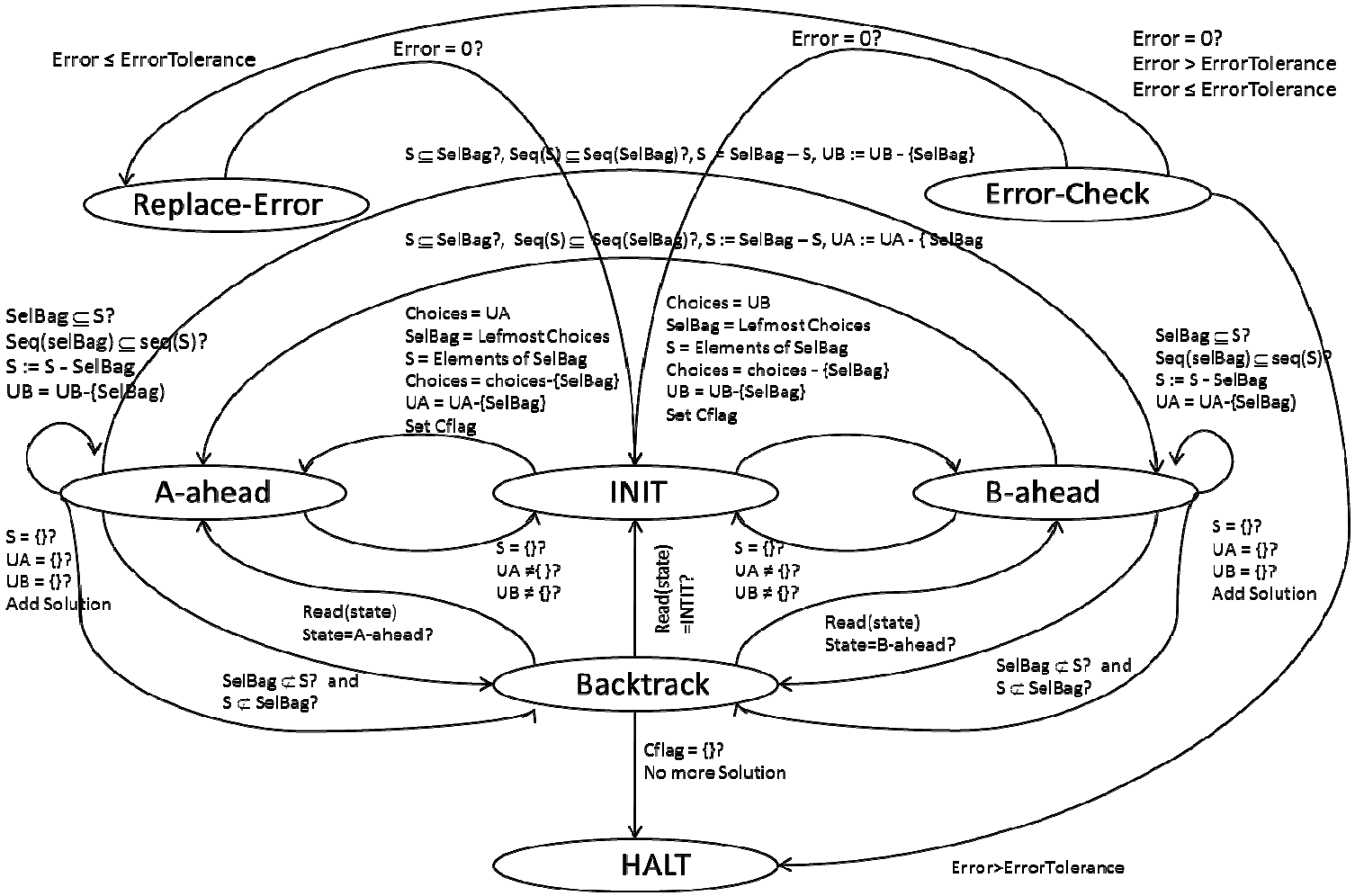

Figure 2-4: Constraint Automaton Version 3

Let's consider the sequence of plasmid puc57. If we use the procedure discussed in section 2.2 to collect fingerprints of the sequence, we will get the following fingerprint data:

Big-Bag-A	Big-Bag-B
355	355 19
19 196	196 288 121 13 541
288	416
121	373
13	320
541 416 373 320 68	68

Figure 2-5: Fingerprints of puc57 without errors

While measuring the length of each sub-fragment, if we mismeasured the fragments, and end up getting "17" instead of "13" in *Big-Bag-A* and "418" instead of "416" in *Big-Bag-B*, then we will have the following as input data.

Big-Bag-A	Big-Bag-B
355	355 19
19 196	196 288 121 **13** 541
288	**418**
121	373
17	320
541 **416** 373 320 68	68

Figure 2-6: Fingerprints of puc57 with errors

The automaton compares each element of both bags and finds that 17 and 13, and 416 and 418 do not match. But it does not have any idea about which one of the data elements is the right one. So it takes the mean of two mismatched elements and replaces the wrong data with the mean in both *Big-Bag-A* and *Big-Bag-B*.

Big-Bag-A	**Big-Bag-B**
355	355 19
19 196	196 288 121 **15** 541
288	**417**
121	373
15	320
541 **417** 373 320 68	68

Figure 2-7: Wrong data is replaced with mean of two mismatched data

After replacing the wrong data, both bags contain the same set of elements and they move to INIT state.

3. The automaton comes to INIT state, if either the data has no error or if all the errors have been replaced by the means of two mismatched data. All the A bags are stored in *UA* and all B bags are stored in *UB*. Since this is the first state where it actually starts processing data for solving GMAP, it can pick any bag from either the big bag A or B. So *Options* is set to all the bags from A and B. The leftmost bag is selected from *Options* and set to *SelBag* and remaining bag is set to *Choices*. The elements of *SelBag* are set to S. Since *Choices* is not empty, *Cflag* is set to 1. The bag which is just selected is removed from either *UA* or *UB,* from where ever it belongs The elements of the bag in *SelBag* are set to either Length-A or Length-B. If the Length-A

is greater than Length-B, the automaton goes to A-ahead state, if not then it moves to B-ahead state.

Using figure 2.5, if A1 is picked as the *SelBag*, then, UA=A2, A3, A4, A5; S = 355 and Length-A = 355. The automaton goes to A-ahead state.

4. If the automaton is in A-ahead state, the next bag to be picked is selected from *UB*. The automaton tries to match the elements of S and *SelBag* as far as possible.

   - If S ⊆ *SelBag*, then the automaton moves to B-ahead state and S is set to the difference between S and the elements of *SelBag*.

   - If *SelBag* ⊆ *S*, then the automaton moves to A-ahead state and S is set to the difference between S and the elements of *SelBag*

   The elements of S are now compared with the elements of all the bags of *UB*. All the bags that are either a subset or a superset of S are set to *Options*. If it doesn't find any bags that meet these criteria, the automaton goes to Backtrack state. Using figure 2.5, if S = "541, 288, 121, 373, 68, 416, 320" and *Options* = "B3, B4, B6, B5". Again, leftmost bag is picked and set to *SelBag*.

5. If the automaton is in B-ahead state, the next bag is selected from *UA*. Similar procedure as in step 4 is followed to select the next *SelBag*. Again, if *SelBag* is empty, it goes to backtrack.

6. If the automaton is in the backtrack state, it goes back to the last node of which *Cflag* is set to 1. The idea is to select some other bags from *Options* besides the one picked

initially, which might have led to the backtrack state. So it tries to go to a node where there are some other Choices. For this reason *Cflag* was used initially to help keep track of Choices, if we need to come back and look for other options.

It retrieves the values of all state variables from that particular node and resets the current values of state variables with the one from that node. It either goes to A-ahead or B-ahead state, depending on the current value of Length-A or Length-B. No matter which state it goes to, it picks the next bag from *Options*, discarding the ones which it has already tried and has failed.

7. If Length-A is greater than Length-B, automaton goes to A-ahead state; if Length-B is greater it goes to B-ahead state. But sometimes, both Length-A and Length-B can be equal. In this case, the automaton selects one random bag from *UA* and continues the automaton.

8. If all the bags in *UA* and *UB* are used and if S is empty, the solution has been found and it goes to HALT state.

Chapter 3

3. Methodology

The input for the implemented constraint automaton is the two bags; *Big-Bag-A* and *Big-Bag-B* obtained by following Revesz's Fingerprint described in section 2.2. These bags are then fed to the constraint automaton to find the right order of individual fragments.

The constraint automaton was implemented using Perl v5.12.2 (Practical Extraction and Report Language). [1] The program was compiled in Eclipse SDK v3.5.2 [7] using EPIC (Eclipse Perl Integration). Eclipse is a multi-language software development environment. EPIC is an open source Perl Integrated Development Environment is based on the Eclipse platform.

3.1 Data Sources and Data Collection

To implement and test any algorithm we need to have data sets. The data for the implemented algorithm are the sequences of DNA of plasmids and phage. The sequence of DNA is often stored in a flat text file called FASTA file. It is a text-based format for representing nucleotide sequence or peptide sequence. "A sequence in FASTA format begins with a single-line description, followed by lines of sequence data. The description line is distinguished from the sequence data by a greater than (">") symbol in the first

column". [20] The FASTA files for the plasmids and phage were downloaded from New England BioLabs. [20]

One way of making copies of a DNA is to insert a DNA piece into the genome of an organism, a *host* or *vector* and let the organism multiply itself. The inserted piece (the *insert*) gets multiplied along with the original DNA of the host upon host multiplication.

"A plasmid is a piece of circular DNA that exists in bacteria." [30]. It replicates itself when the cell divides and each copy of a daughter cell keeps one copy of the plasmid. Plasmids make good vector but can only handle inserts up to 15kbp [30].

Bacteriophages or just phages are viruses that infect bacteria. They are often used as vectors. Inserts in phage DNA get replicated when the virus infects a host bacterium.

To observe the variation in computational complexity with respect to different length sequences, plasmid and phage ranging from 2710 to 35937 bp were chosen.

The following plasmids and phage are used to test the purposed algorithm.

1. pUC57: 2710 base pair
2. pTXB1: 6706 base pair
3. pKLAC-malE – 10153 base pair
4. pB85766 – 14875 base pair
5. Adenovirus-2 – 35937 base pair

To collect fingerprints of sequences as in section 2.2, restriction enzymes MvaI and MaeII were used. MvaI is an isolate from *Micrococcus varians RFL19* and has restriction site at CC^WGG. "W" can be either A or T. MaeII is isolated from *Methanococcus aeolicus* and has restriction site at A^TCG.

DNA sequences were cleaved into fragments and sub-fragments by using Webcutter 2.0 [35] See table 3.1 and 3.2 and appendix AA-AC for the fingerprints of pUC57, pTXB1, pKLAC-ma1E, pB85766 and Adenovirus respectively.

Each DNA sequence is first cleaved by MvaI and each individual fragement was again cleaved by MaeII to obtain sub-fragments. This is the data for Big-Bag-A. Again, each DNA sequence is cleaved by MaeII and then by MvaI to obtain *Big-Bag-B*.

No	Fragments	Sub-Fragments
1	355	355
2	215	19, 196
3	288	288
4	121	121
5	13	13
6	1709	541, 416, 373, 320, 68

Table 3-1:Big-Bag-A for pUC57

No	Fragments	Sub-Fragments
1	374	355 19
2	1159	196 288 121 13 541
3	416	416
4	373	373
5	320	320
6	68	68

Table 3-2:Big-Bag-B for pUC57

3.2 Implementation of Constraint Automaton for GMAP

The web based version of the program is hosted in the web server of Department of Computer Science and Engineering, UNL.

The program takes a text file which contains the elements for both *Big-Bag-A* and *Big-Bag-B*. Each big bag has to start with the '<BAG>' tag and each bag starts in a new line. Elements within each bag are separated with a space.

Figure 3-1: Example of Input File

The program is hosted in http://cse.unl.edu/~singh/gmap.htm. The input file is selected from the local machine and submitted to the automaton by clicking the submit button. The automaton then compares the elements of one big bag with the other and checks if there are elements without the matching pairs in the other bag. If the elements match, the automaton moves to the next state. If not then the automaton checks whether or not the difference between two unmatched elements is less than the specified threshold or not.

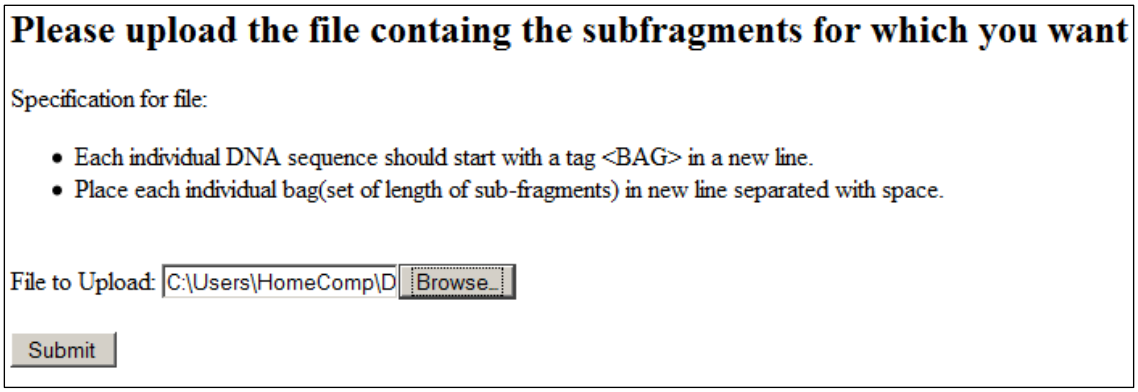

Figure 3-2: Submission of Input File

Let's consider three different scenarios:

1. Input data has errors greater than the specified threshold value
2. Input data has errors less than or equal to the threshold value
3. Input data does not have any errors

1. Input data has errors greater than the specified threshold value

If the error is greater than the specified threshold value, the program lets the user know about it and asks the user to reenter the data.

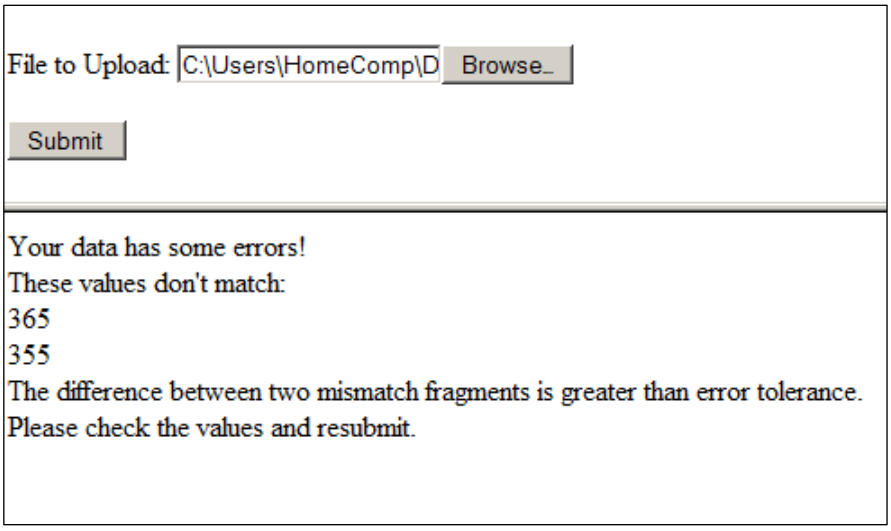

Figure 3-3: If error is greater than specified threshold

2. Input data has errors less than or equal to the threshold value

If the error, which is the difference in length between two unmatched elements is less than or equal to the threshold then the automaton calculates the mean of two mismatched elements and substitutes the wrong value with the mean. In this example we consider threshold value to be 5.

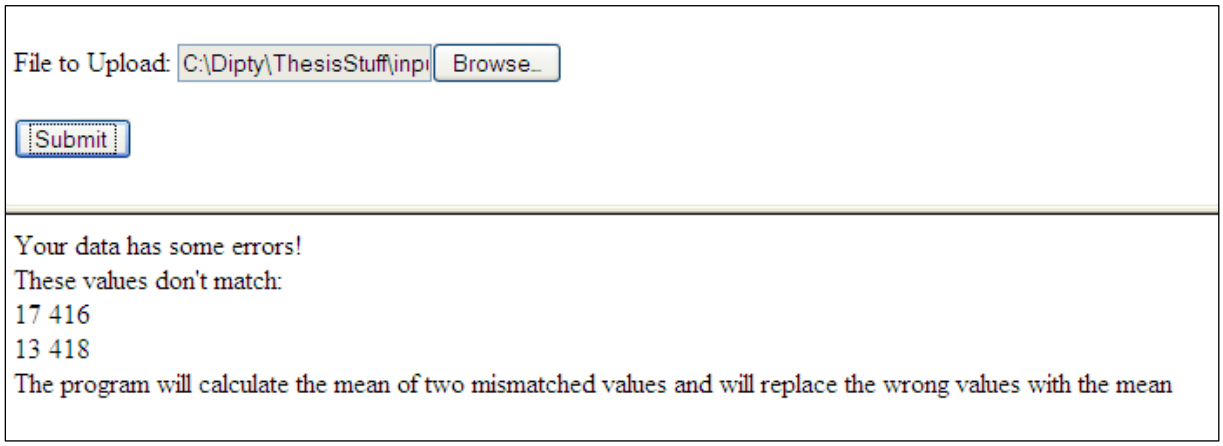

Figure 3-4: If error is less than or equal to the threshold

Considering the data in figure 2-2 as an input data the program will replace the wrong data with the mean of two mismatched data. The program stores all the elements of *Big-Bag-A* in bags A (A1, A2, A3…An) and all the elements of *Big-Bag-B* in bags B (B1, B2, B3…Bn)

Original Input	**Randomized Input Data with Error**	**Bags**
<BAG>	<BAG>	Big-Bag-
355	**355**	A1
19 196	19 196	A2
288	**288**	A3
121	121	A4
13	541 416 373 **320** 68	A5
541 416 373 320 68	13	A6
<BAG>	<BAG>	Big-Bag-
355 19	**359** 19	B1
196 288 121 13 541	416	B2
416	373	B3
373	**325**	B4
320	68	B5
68	196 **285** 121 13 541	B6

Table 3-3: Input data with error which is less than or equal to threshold value

The program reconstructs the original DNA based on overlaps on sub-fragments of *Big-Bag-A* and *Big-Bag-B*

3. Input data doesn't have any error

If the input data doesn't have any error, the program goes directly to the fragment assembling process without having to go through any error substitutions.

Chapter 4

4. Results and Analysis

Input data of all five DNA sequences were fed to the implemented application and the results were analyzed separately and also compared with one another. The results were compared by the time it takes to assemble each subsequence for both erroneous and error free data. To better analyze the results three different scenarios were considered.

4.1 Input data without error

Input data for all five DNA sequences; pUC57, pTXB1, pKLAC-malE, pB85766 and Adenovirus-2, were fed to the application. The program was executed 10 times for each DNA sequence and the average execution time was noted. The data was randomized to portrait the real scenario of wet lab experiments and fed to the automaton. The fingerprints of fragments taken from the input table 3.1 and 3.2, and appendix AA-AD, are used to assemble the DNA sequences of plasmids and phage. The original order of the sequence and the one assembled by the program were then compared.

355 19 196 13 288 121 541 416 373 320 68

Figure 4-1: Original order of subfragments of pUC57

Overlapping fragments were used to reconstruct the original DNA. Adjacent grey nodes were then permuted to obtain the original DNA sequence.

A1	A2	A3	A4	A5	A6
355	19 196	288	121	541 416 373 320 68	13
355 19	196 13 288 121 541	416	373	320	68
B1	B6	B2	B3	B4	B5

Table 4-1: Fragment Assembly of pUC57 done by Application

Node No	**CurrBag**	**S**	**UA**	**UB**	**Options**	**SelBag**	**Choices**	***Cflag***
1	A1	{355}	A2 A3 A4 A5 A6	B1 B2 B3 B4 B5 B6	{B1}	{B1}	{}	0
2	B1	{19}	A2 A3 A4 A5 A6	B2 B3 B4 B5 B6	{A2}	{A2}	{}	0
3	A2	{196}	A3 A4 A5 A6	B2 B3 B4 B5 B6	{B6}	{B6}	{}	0
4	B6	{541 288 15 121}	A3 A4 A5 A6	B2 B3 B4 B5	{A3 A4 A5 A6}	{A3}	{A4 A5 A6}	1
5	A3	{541 15 121}	A4 A5 A6	B2 B3 B4 B5	{A4 A5 A6}	{A4}	{A5 A6}	1
6	A4	{541 15}	A5 A6	B2 B3 B4 B5	{A5 A6}	{A5}	{A6}	1
7	A5	{373 68 417 15 320}	A6	B2 B3 B4 B5	{B2 B3 B4 B5}	{B2}	{B3 B4 B5}	1
8	B2	{373 68 15 320}	A6	B3 B4 B5	{B3 B4 B5}	{B3}	{B4 B5}	1
9	B3	{68 15 320}	A6	B4 B5	{B4 B5}	{B4}	{B5}	1
10	B4	{68 15}	A6	B5	{B4 B5}	{A6}	{B5}	1
11	A6	{68}		B5	{B5}	{B5}	{}	0

Table 4-2: Stepwise fragment assembly of pUC57 without error

Fragment assembly was done in a similar fashion for all the remaining sequences. Input data were used from appendix AA to AD

Table 4.3 shows average execution time for each DNA sequence

DNA	Length(bp)	No of bags	Average Execution Time
pUC57	2710	10	0.31
pTXB1	6706	44	0.84
pKLAC-	10153	46	1.01
pB85766	14875	84	18.1
Adenovirus	35937	221	74.6

Table 4-3: Average Execution Time for error free data

To investigate if the time complexity of the implemented constraint automaton solution is linear, cubic polynomial, quadratic polynomial or exponential, the relationship between the length of DNA sequence and the execution time was examined.

All the functions were plotted using SPSS [18] and the nature of fit was determined by the value of variance (R-square) and P-value. With error free data, the R-square for linear, cubic polynomial, quadratic polynomial and exponential are 0.96, 0.99, 0.98 and 0.84 respectively. This states that 96%, 99%, 98% and 84%of the variance in the execution time can be explained by the length of DNA sequence in linear, cubic, quadratic and exponential equation respectively. The P-values for linear, quadratic polynomial, cubic polynomial, and exponential are 0.003, 0.012, 0.069, and 0.026 respectively. P-value for linear model is less compared with quadratic polynomial, cubic polynomial, and exponential equation. This concludes that the linear function better fits (R-square value approximately equal to 1 and P-value <0.01) execution time and the length of DNA sequence than cubic, quadratic and exponential function.

.

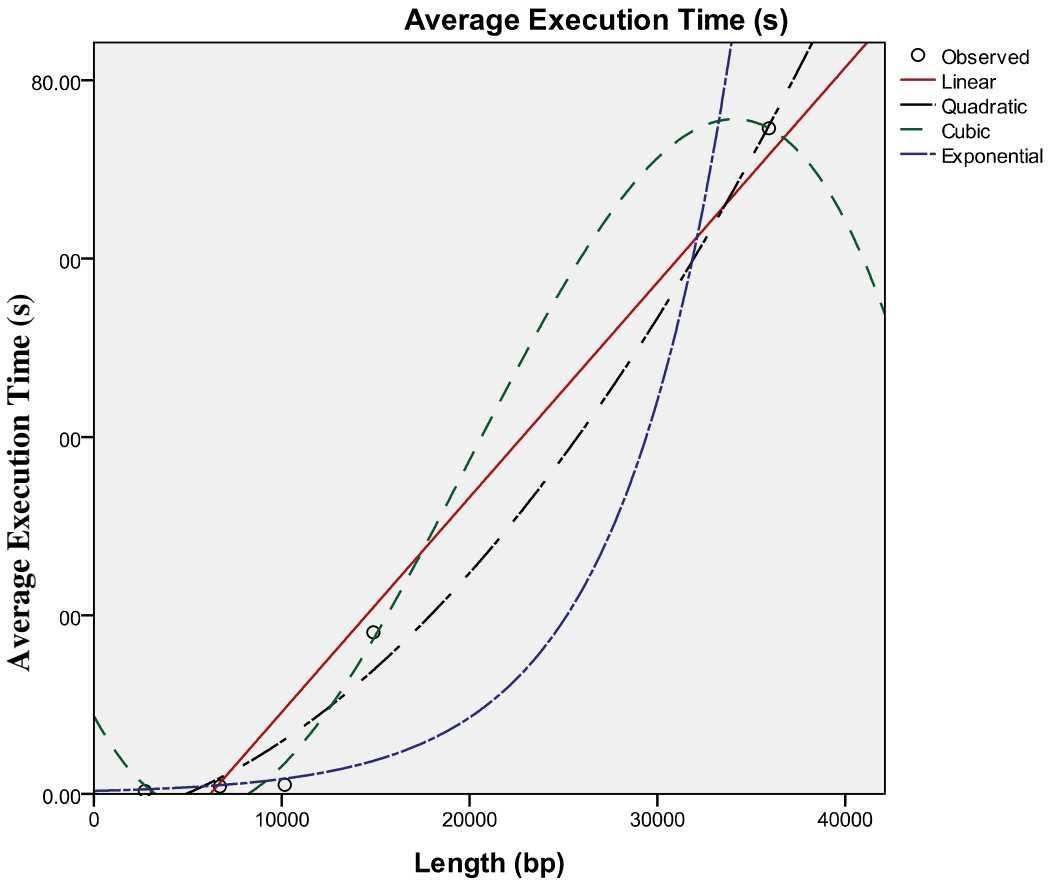

Figure 4-2: Regression Model for Error free data, using Execution time as dependent variable and Length of Sequence as predictor

Functions	P-Value	R^2- Value
Linear	0.96	0.003
Quadratic	0.988	0.012
Cubic	0.997	0.069
Exponential	0.849	0.026

Table 4-4: P and R2 values error free data

4.2 Input data with error

A few data points from the input data (table 3.9) were manipulated to create erroneous data like in the real situation in wet lab. The error tolerance was set to 5.

A1	A2	A3	A6	A5	A4
357	19 196	286	13	541 416 373 322 68	416
357 19	196 13 286 121 541	373	373	320	416
B1	B6	B3	B5	B4	B2

Table 4-5: Fragment Assembly of erroneous fragments of pUC57

Node No	CurrBag	S	UA	UB	Options	SelBag	Choices	*Cflag*
1	A1	{357}	A2 A3 A4 A5 A6	B1 B2 B3 B4 B5 B6	{B1}	{B1}	{}	0
2	B1	{19}	A2 A3 A4 A5 A6	B2 B3 B4 B5 B6	{A2}	{A2}	{}	0
3	A2	{196}	A3 A4 A5 A6	B2 B3 B4 B5 B6	{B6}	{B6}	{}	0
4	B6	{286 13 541 121}	A3 A4 A5 A6	B2 B3 B4 B5	{A3 A6 A5 A4}	{A3}	{A6 A5 A4}	1
5	A3	{13 541 121}	A4 A5 A6	B2 B3 B4 B5	{A6 A5 A4}	{A6}	{A5 A4}	1
6	A6	{541 121}	A4 A5	B2 B3 B4 B5	{A5 A4}	{A5}	{A4}	1
7	A5	{373 68 322 121 416}	A4	B2 B3 B4 B5	{B3 B5 B4 B2}	{B3}	{B5 B4 B2}	1
8	B3	{68 322 121 416}	A4	B2 B4 B5	{B5 B4 B2}	{B5}	{B4 B2}	1
9	B5	{322 121 416}	A4	B2 B4	{B4 B2}	{B4}	{B2}	1
10	B4	{121 416}	A4	B2	{B4 B2}	{A4}	{B2}	1
11	A4	{416}		B2	{B2}	{B2}	{}	0

Table 4-6: Stepwise fragment assembly of pUC57 with error

The average execution time for erroneous input data for each DNA sequences is as follows:

DNA Sequence	Length(bp)	No of bags	Average Execution Time (s)
pUC57	2710	10	0.32
pTXB1	6706	44	0.88
pKLAC-malE	10153	46	1.04
pB85766	14875	84	19
Adenovirus	35937	221	76

Table 4-7: Average Execution time with erroneous data

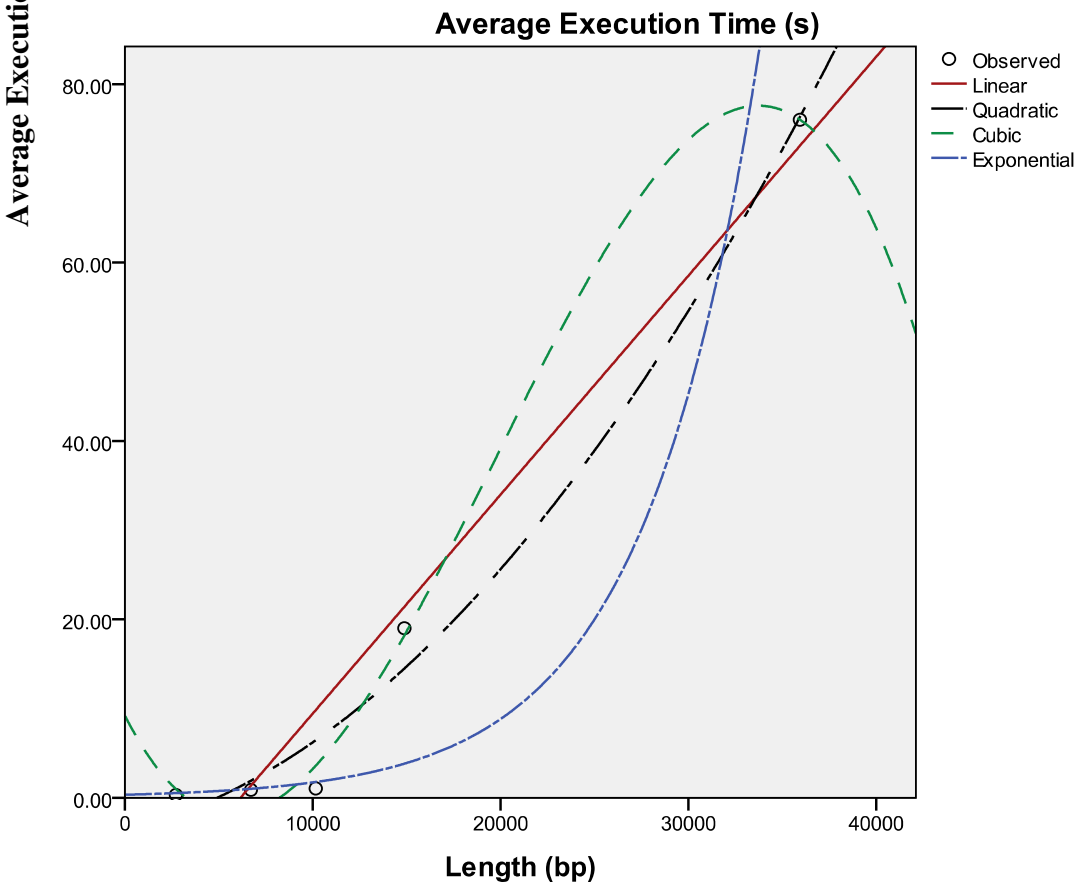

Figure 4 3: Regression Model for Erroneous data, using Execution time as dependent variable and Length of Sequence as predictor

Functions	P-Value	R^2- Value
Linear	0.96	0.003
Quadratic	0.987	0.013
Cubic	0.997	0.071
Exponential	0.845	0.027

Table 4-8: P and R2 values error free prone data

With erroneous input data, the R-square value for linear equation is 0.96 and P-value is 0.003. This concludes that the linear function better fits (R-square value approximately equal to 1 and P-value <0.01) execution time and the length of DNA sequence than cubic, quadratic and exponential functions.

4.3 Comparison between error free and error-prone data

Execution times to assembly fragments with and without error were compared. The R-square values for liner function for both of them were found to be similar.

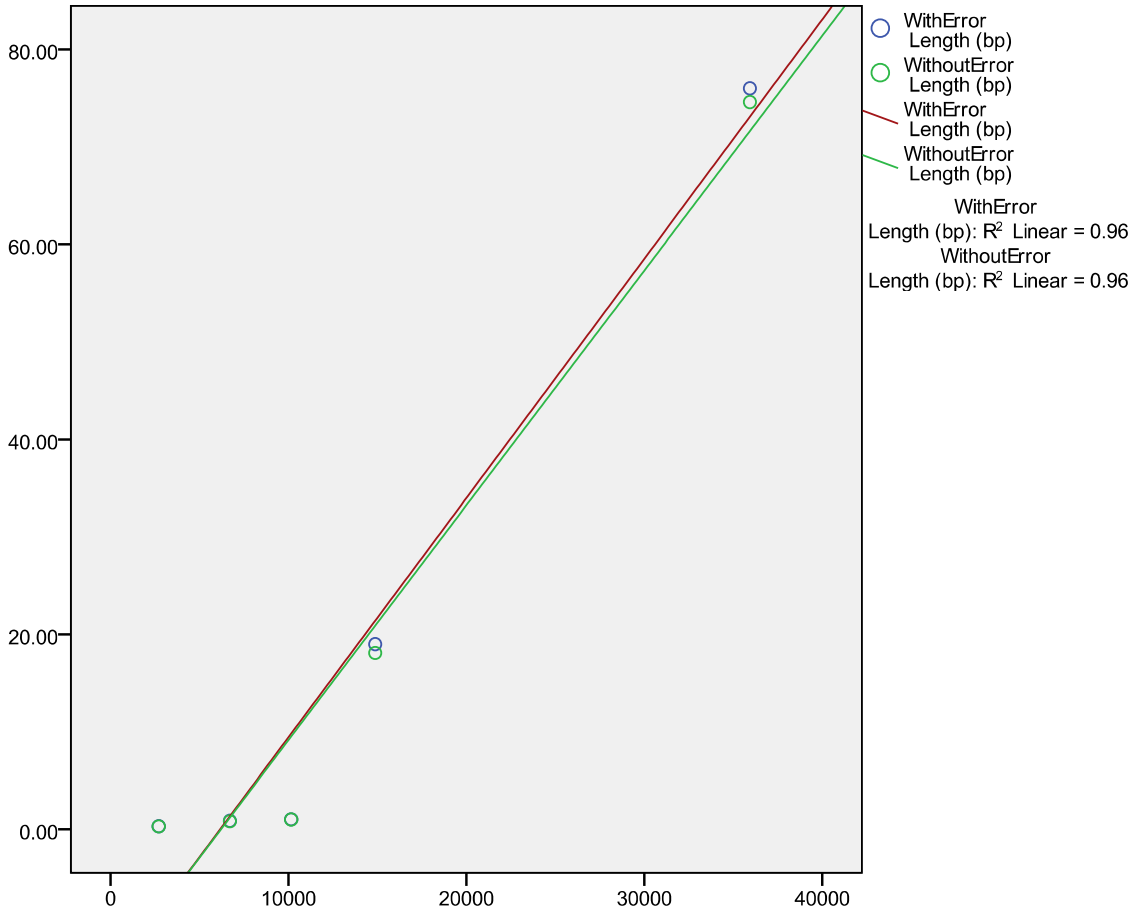

Figure 4-3: Regression Model for precise and error-prone data

4.4 Reduction of Execution time

The execution times for different sets of input were analyzed to predict the pattern of output and use the results to help reduce the execution time to find the solution for GMAP.

DNA Sequence of:	No of A-bags	No of B-bags	Difference |A-B|	AverageExecution Time (s)
pUC57	5	5	0	0.31
pTXB1	20	24	4	0.84
pKLAC-malE	21	25	4	1.01
pB85766	51	33	18	18.1
Adenovirus	137	84	53	74.6

Table 4-9: Number of bags when restriction enzymes MaeII and MvaI are used

From table 4-6; the difference between number of bags in *Big-Bag-A* and *Big-Bag-B* for pB85766 is 18. The higher the difference between numbers of bags, there is a high chance that there will be more bags with just one element in which ever big bag has higher number of bags.

38; 8; 507; 130; 235; 24; 8; 8; 53; 8; 8; 59

Figure 4-4: Number of bags of pB85766 with single element

If the numbers of bags with single element is high, there is less likelihood of finding the overlapping fragments in the opposite Big-Bag without backtracking multiple times. The output file in Appendix BA shows the stepwise fragment assembly for pB85766. The program backtracks several times and ultimately finds the solution at an average execution time of 18.1 seconds.

To investigate whether or not difference in numbers of bags with two big bags effect the execution time, another pair of restriction enzymes; *MaeI*(C^TAG) and *HinfI*(G^ANTC)

were applied to the DNA sequence of pB85766 . The resulted input data is listed in Appendix AE-1 and AE-2.

DNA Sequence of:	No of A-bags	No of B-bags	Difference |A-B|	AverageExecution Time (s)
pB85766	44	40	4	3.5

Table 4-10: Number of Bags when MaeI and HinfI are applied to pB85766

Even though the total number of bags in table 4-6 and table 4-7 are almost the same, the difference between numbers of bags is significantly reduced. This resulted in many overlapping fragments in opposite big bag and led to finding the solution within 3.5 seconds without many backtrackings (Appendix BB)

4.4 Estimation of Error tolerance value

Varieties of erroneous input data with different error limit were tested to estimate the error tolerance of the implemented algorithm. To find errors in input data, the algorithm tries to match the sub-fragments of two big bags. If it finds any mismatching data, it concludes that the input data has an error.

Original Input Data	Input Data With Error
<BAG> 355 19 196 288 121 541 416 373 320 68 13 <BAG> 355 19 416 373 320 68 196 288 121 13 541	<BAG> 355 10 196 288 121 541 416 373 320 68 13 <BAG> 355 19 416 373 320 68 196 288 121 3 541

Table 4-11: Fingerprints of pUC57 with error tolerance set to 10

In table 4.6, the original fingerprint “19” is replaced with “10” and “13” with 3. When the algorithm finds that the input data has some error, it tries to find the closest match of the error data among the error data in the other big bag. In this example, instead of matching “10” with the correct fingerprint “19”, it picks the closest match “3” and the same with “13” and “19”

Error tolerance was set to 5 and changes were made in table 4.10 accordingly. When this input data was fed to the algorithm, “13” was matched with“8” and also “14” and “19” were matched, the way it is supposed to be.

Original Input Data	Input Data With Error
<BAG> 355 **19** 196 288 121 541 416 373 320 68 13 <BAG> 355 19 416 373 320 68 196 288 121 **13** 541	<BAG> 355 **14** 196 288 121 541 416 373 320 68 13 <BAG> 355 19 416 373 320 68 196 288 121 **8** 541

Table 4-12: Fingerprints of pUC57 with error. Tolerance set to 5

Many other input data were tested to find the error tolerance of the algorithm. The results let to the conclusion that, the minimum error tolerance level depends on the input data. The higher the lowest difference between the lengths of sub-fragment, the higher is the error tolerance.

Fingerprints of Big Bag A	Fingerprints of Big Bag B	Difference
19	13	6
68	19	49
121	68	53
196	121	75
288	196	92
320	288	32
355	320	35
373	355	18
416	373	43
541	416	125

Table 4-13: Calculation of Error tolerance of puc57

Minimum Error Tolerance for puc57 = Lowest difference between two fragments -1

= 6-1 = 5

Maximum Error Tolerance for puc57 = Highest difference between two fragments -1

= 125-1 = 124

As long as the erroneous does not interfere with the other elements of the data set, it can tolerate error anywhere between the minimum and maximum error tolerance value.

Chapter 5

5. Conclusion and Future Work

5.1 Conclusion

The Genome Map Assembly Problem was solved using an abstraction of the GMAP, the *Big-Bag Matching Problem,* and the constraint automaton solution proposed by Revesz [25]. The solution was further modified by Revesz and Ramanathan.[24] which incorporated backtracking and further proving that the execution time to solve the GMAP using Constraint Automaton Solution is linear This thesis modifies the existing constraint automaton solution to make it find the solution when errors were introduced.

The input data to test the implemented application were the fingerprints of DNA sequence collected from New England BioLabs [20]. Restriction enzymes MaeII and Mval were used to cleave the DNA sequences at certain length and create the fingerprints for each DNA sequences.

The relationship between the length of input DNA sequences, and the execution time to assembly the sub-fragments of those DNA were examined for both error-prone and error-free data using curve estimation in regression modeling. R-square value and P-value indicated that a linear function better fits execution time and the length of DNA sequence than cubic, quadratic and exponential functions.

Different threshold values were tested to estimate the minimum error tolerance value for the implemented application. The application tolerates the threshold up to the lowest value (greater than 0) of the differences between the lengths of each sub-fragment.

5.2 Future Work

The input data for this algorithm are the sets of fingerprints of sub-fragments. Since fingerprints are the lengths of sub-fragments, the one with the same length will end up having the same fingerprint even though they might not necessarily have same sequences.

Fingerprint	Sequence
81	agggccacatccttggccaattgcaagccatcaacaaagcccgccaagagtttctgctacgaaagggacgg ggggtttacc
81	tggacccccagtccggcgagggctcaacccaatccccccgccgccgcagccctatcagcagccgcgggc ccttgcttccc

Table 5-5-1: Sub-Fragment from Adenovirus2 with same fingerprint but different sequences

During fragment assembly the algorithm does not distinguish between these two sub-fragments and may result in wrong assembly of original DNA In the future, this application can be modified to create unique fingerprint for each sub-fragment, so that the fragment assembly is more precise even in the presence of duplicate length sub-fragments.

Bibliography

1. Active Perl. 12 6 2010 <http://www.activestate.com/activeperl/downloads>.
2. Arber, W and S Linn. "DNA modification and restriction." 38 (1969): 467-500 .
3. Bickle, T A and D H Kruger. "Biology of DNA Restriction." (1993): 434-450.
4. Brody, T. Nutritional Biochemistry. Academic Press, 1994.
5. California State University, Sacramento. 24 2 2011 <http://www.csnu.edu>.
6. Coulson, A, et al. "Toward a physical map of the genome of the nematode. Caenorhabditis elegans." 83 (1986): 7821-7825.
7. Eclipse. 01 06 2010 <http://www.eclipse.org/downloads/>.
8. Fasulo, D P, et al. "An Algorithmc Approach to Multiple Complete Digest Mapping." (1997).
9. Genetics Home Reference. 20 2 2011 <http://ghr.nlm.nih.gov/>.
10. Genome News Network. <http://www.genomenewsnetwork.org>.
11. Genome Program of the U.S. Department of Energy. 03 02 2011 <http://genomics.energy.gov/>.
12. Gillett, W, et al. "Assembly of High-Resolution Restriction Maps Based on Multiple Complete Digests of a Redundant Set of Overlapping Clones." (1996).
13. Green, E and P Green. "Sequence-tagged site (STS) content mapping of human chromosomes: theoretical considerations and early experiences." 2 (1991): 77-90.
14. Harley, E and A J Bonner. "A Flexible Approach to Genome Map Assembly." (1994).
15. Jamison, D C. Perl Programming for Biologists. New Jersery: John Wiley and Sons, 2003.
16. Karp, R M. "Mapping the Genome: Some Combinatorial Problems Arising in Molecular Biology." (1993): 278-285.

17. Lander, E S and M L Waterman. "Genomic Mapping by Fingerprinting Random." 2 (1988): 231-239.

18. Levesque, Raynald. Raynald's SPSS Tools. 15 2 2011 <http://www.spsstools.net/>.

19. McGraw Hill Higher Education. 15 01 2011 <http://catalogs.mhhe.com>.

20. New England BioLabs. 10 01 2011 <http://www.neb.com>.

21. Oak Ridge National Laboratory. 25 02 2011 <http://www.ornl.gov/>.

22. Olson, M V, et al. "Random-clone strategy for genomic restriction mapping in yeast." 83 (1986): 7826-7830.

23. Pevzner, P A. Computational Molecular Biology: an Algorithmic Approach. 2000.

24. Ramanathan, V and P Z Revesz. "Constraint Database Solutions to the Genome Map Assembly Problem." Paris, France: Spinger, 2004. 88-111.

25. Revesz, P Z. "Bioinformatics." Introduction to Constraint Databases. New York: Springer, 2002. 351-360.

26. Revesz, P Z. "Constraint Automata." Introduction to Constraint Databases. New York: Springer, 2002. 62-78.

27. Revesz, P Z. "Refining Restriction Enzymes Genome Maps." 1997. 361-375.

28. Revesz, Peter. Introduction to Databases: From Biological to Spatio-Temporal. New York: Springer, 2010.

29. Setubal, C and J Meidanis. "Fragment Assembly of DNA." Introduction to Computational Molecular Biology. Pacific Groce, CA: Brooks/Cole Publishing Company, 1997. 118-124.

30. Setubal, C and J Meidanis. "Basic Concepts of Molecular Biology." Introduction to Computational Molecule Biology. Pacific Grove, CA: Brooks/Cole Publishing Company, 1997. 1-30.

31. Siegel, A F, et al. "Optimization of Restriction Fragment DNA Mapping." 5 (1998): 113-126.

32. Tisdall, James. Beginning Perl for Bioinformatics. Sebastopol, CA: O'Reilly Media, 2001.

33. Tisdall, James. Mastering Perl for Bioinformatics. Sebastopol, CA: O'Reilly Media, 2003.

34. Understanding DNA and its Relevance to Life at Explore DNA. 02 01 2011 <http://www.exploredna.co.uk>.

35. Webcutter. 01 01 2011 <http://bio.lundberg.gu.se/cutter2/>.

36. Willett, E. "Genomes-Reading the Genetic Code." Genetics Demystifiled. 53-66.

37. Wolfram MathWorld: The Web's Most Extensive Mathematics Resource. 05 01 2010 <http://mathworld.wolfram.com/>.

38. Wong, G K, et al. "Multiple-complete-digest restriction fragment mapping:Generating sequence-ready maps for large-scale DNA sequencing." (1997): 5225-5230.

Appendix

A. Input Tables

AA. pTXB1

No.	Fragments	Big Bag A pTXB1
1	2151	5, 320, 373, 411, 159, 12,
2	13	13
3	121	121
4	1179	215, 426, 88, 56, 394
5	57	57
6	540	540
7	315	315
8	301	62, 239
9	168	113, 55
0	15	15
11	168	133, 55
12	15	15
13	168	133, 15
14	15	15
15	168	133, 15
16	15	15
17	168	133, 15
18	599	443, 156
19	234	234
20	296	68, 52, 176

Table A-1:Big-Bag-A for pTXB1

No	Fragment	Big Bag B pTXB1
1	5	5
2	320	320
3	373	373
4	411	411
5	159	159
6	12	12
7	43	43
8	110	110
9	176	176
10	891	542, 13, 121, 215
11	426	426
12	88	88
13	56	56
14	1295	394, 57, 540, 304
15	73	11, 62
16	352	239, 113
17	183	55, 15, 113
18	183	55, 15, 113
19	183	55, 15, 113
20	183	55, 15, 113
21	498	55, 443
22	458	156, 234, 68
23	52	52
24	176	176

Table A-2:Big-Bag-B for pTXB1

AB. pKLAC-malE

No	Fragments	Big-Bag-A for pKLAC-malE
1	562	527,23,12
2	248	93,87,68
3	406	44,362
4	312	312
5	471	471
6	1265	771,494
7	630	394,236
8	455	455
9	243	155,88
10	82	82
11	630	630
12	354	111,243
13	1132	169,288,121,13,541
14	416	416
15	373	373
16	439	439
17	110	110
18	43	43
19	12	12
20	307	288,19
21	1116	366,12.287,5,446
22	15	15
23	49	18,8,,23
24	189	189
25	294	294

Table A-3:Big-Bag-A for pKLAC-malE

No.	Fragments	Big-Bag-B for pKLAC-malE
1	527	527
2	23	23
3	105	12, 93
4	87	87
5	112	68, 44
6	1916	362, 312, 471, 771
7	888	494, 394
8	846	236, 455, 155
9	911	88, 82, 630, 111
10	412	243, 169
11	288	288
12	121	121
13	13	13
14	2222	541, 416, 373, 439, 110, 43, 12, 288
15	385	19, 366
16	12	12
17	287	287
18	5	5
19	479	446, 15, 18
20	8	8
21	506	23, 189, 294

Table A-4:Big-Bag-B for pKLAC-malE

AC. Adenovirus2

No.	Fragments	Big Bag A for Adenovirus
1	389	65, 32, 74, 203, 15
2	1073	318, 755
3	71	71
4	283	283
5	102	102
6	254	254
7	49	49
8	282	282
9	263	263
10	68	68
11	138	138
12	183	158, 25
13	204	204
14	797	797
15	177	4, 173
16	294	57, 237
17	45	28, 17
18	80	80
19	406	406
20	119	22, 97
21	87	87
22	66	66
23	199	199
24	32	32
25	184	184
26	68	68
27	249	67, 182
28	240	181, 59
29	648	334, 69, 132, 93, 20
30	156	156
31	291	291
32	689	588, 101
33	256	256
34	18	18
35	362	186, 176
36	78	19, 59
37	994	169, 103, 722
38	197	197
39	228	205, 23
40	121	121
41	9	9
42	282	282
43	49	10, 39
44	437	437
45	24	24
46	264	73, 174, 17
47	30	30
48	285	34, 251
49	162	82, 80
50	99	99
51	87	87
52	22	22
53	202	33, 169
54	192	192
55	63	49, 14
56	69	69
57	198	49, 149
58	219	219
59	51	51
60	304	304
61	512	392, 120
62	44	44
63	212	212
64	577	577
65	48	48
66	71	71
67	30	30
68	45	45
69	558	409, 149
70	237	46, 75, 102, 14
71	179	179
72	327	189, 66, 72
73	6	6
74	238	65, 173
75	149	149
76	398	36, 362
77	271	72, 65, 134
78	217	217
79	567	567
80	76	76
81	30	30
82	9	9
83	298	298
84	302	174, 128
85	204	57, 55, 92
86	276	39, 237
87	215	215
88	200	200
89	828	445, 248, 72, 63
90	344	258, 86
91	228	193, 35
92	36	36
93	327	199, 128
94	319	109, 210
95	88	88

96	307	307
97	27	27
98	21	21
99	493	233, 260
100	71	71
101	542	320, 222
102	838	75, 504, 20, 239
103	62	62
104	666	79, 587
105	231	210, 21
106	15	7, 8
107	36	19, 17
108	125	46, 42, 37
109	34	34
110	96	96
111	321	256, 65
112	234	120, 114
113	81	81
114	81	81
115	309	309
116	902	619, 283
117	9	9
118	45	45
119	552	552
120	109	14, 95
121	176	176
122	686	138, 288, 260
123	291	291
124	31	31
125	43	43
126	829	4, 825
127	943	943
128	546	546
129	997	163, 834
130	67	67
131	949	41, 677, 231
132	117	63, 54
133	259	57, 9, 166, 27
134	95	95
135	10	10
136	655	307, 12, 336
137	1082	800, 117, 66, 32, 67

Table A-5: Big-Bag-A for Adenovirus

No.	Fragments	Big Bag B for Adenovirus
1	65	65
2	32	32
3	74	74
4	203	203
5	333	15, 318
6	2423	755, 71, 283, 102, 254, 49, 282, 263, 68, 138, 158
7	1030	25, 204, 797, 4
8	230	173, 57
9	265	237, 28
10	525	17, 80, 406, 22
11	800	97, 87, 66, 199, 32, 184, 68, 67
12	363	182, 181
13	393	59, 334
14	69	69
15	132	132
16	93	93
17	1055	20, 156, 291, 588
18	561	101, 256, 18, 186
19	195	176, 19
20	228	59, 169
21	103	103
22	1124	722, 197, 205
23	445	23, 121, 9, 282, 10
24	573	39, 437, 24, 73
25	174	174
26	81	17, 30, 34
27	333	251, 82
28	321	80, 99, 87, 22, 33

29	410	169, 192, 49
30	132	14, 69, 49
31	1115	149, 219, 51, 304, 392
32	1556	120, 44, 212, 577, 48, 71, 30, 45, 409
33	195	149, 46
34	75	75
35	102	102
36	382	14, 179, 189
37	66	66
38	143	72, 6, 65
39	358	173, 149, 36
40	434	362, 72
41	65	65
42	1505	134, 217, 567, 76, 30, 9, 298, 174
43	185	128, 57
44	55	55
45	131	92, 39
46	1097	237, 215, 200, 445
47	248	248
48	72	72
49	321	63, 258
50	279	86, 193
51	270	35, 36, 199
52	237	128, 109
53	886	210, 88, 307, 27, 21, 233
54	651	260, 71, 320
55	297	222, 75
56	504	504
57	20	20
58	380	239, 62, 79
59	797	587, 210
60	28	21, 7
61	27	8, 19
62	63	17, 46
63	42	42
64	423	37, 34, 96, 256
65	185	65, 120
66	1204	114, 81, 81, 309, 619
67	903	283, 9, 45, 552, 14
68	409	95, 176, 138
69	288	288
70	629	260, 291, 31, 43, 4
71	2477	825, 943, 546, 163
72	942	834, 67, 41
73	677	677
74	294	231, 63
75	111	54, 57
76	9	9
77	166	166
78	439	27, 95, 10, 307
79	12	12
80	1136	336, 800
81	117	117
82	66	66
83	32	32
84	67	67

Table A-6: Big-Bag-B for Adenovirus

AD. pB85766

No	Fragments	Big Bag A for pB85766
1	527	153, 297,77
2	75	75
3	285	70,215
4	384	384
5	125	125
6	748	418,330
7	330	262,68
8	370	120,152,98
9	285	89,53,143
10	281	281
11	254	190,64
12	396	10,386
13	1318	256,19,392,636,15
14	532	147,385
15	394	11,287,96
16	152	152
17	1179	344,45,390,53,30,317
18	69	41,18
19	182	182
20	1916	1063,840,13
21	68	68
22	174	174
23	44	44
24	109	24,85
25	156	147,9
26	580	31,549
27	376	376
28	262	262
29	122	122
30	896	495, 162,239
31	38	38
32	8	8
33	507	507
34	130	130
35	235	235
36	24	24
37	8	8
38	83	83
39	8	8
40	8	8
41	78	78
42	47	47
43	131	131
44	813	453, 360
45	24	24
46	8	8
47	8	8
48	53	53
49	8	8
50	8	8
51	59	59

Table A-7: Big-Bag-A for pB85766

No	Fragments	Big Bag B for pB85766
1	153	153
2	297	297
3	222	77, 75, 70
4	1142	215,384,125,418
5	592	330,262
6	188	68,120
7	152	152
8	187	98,89
9	53	53
10	614	143,281,190
11	74	64,10
12	642	386,256
13	19	19
14	392	392
15	636	636
16	162	15,147
17	396	385,11
18	287	287
19	592	96,152,344
20	45	45
21	390	390
22	53	53
23	30	30
24	358	317,41
25	1273	181,821,063
26	840	840
27	323	13,68,174,4424
28	232	85,147
29	40	9,31
30	1804	549,376,262,122,495
31	162	162
32	1997	239,38,8,507,130,235,24,8,83, 8,8,78,47,131,453
33	528	36024,8,8,53,8,8,59

Table A-8: Big-Bag-B for pB85766

AE. pB85766 with restriction enzymes MaeI and HinfI

No	Fragments	Big Bag A for pB85766
1	2	2
2	175	9,164
3	223	48
4	646	423
5	913	105,162
6	1189	276
7	1402	213
8	1728	184,86,56
9	1846	118
10	1906	10,50
11	2432	77,58,114,43,190,44
12	2624	6,108,78
13	3227	17,100,421,65
14	3489	262
15	4830	649,171,521
16	5142	312
17	6391	320,41,466,103,319
18	7403	117,235,660
19	7511	108
20	7597	86
21	7821	224
22	9037	305,16,96,218,526,55
23	9373	55,107,174
24	9409	36
25	9814	209,196
26	10075	261
27	10585	510
28	10907	322
29	11195	288
30	11345	150
31	11426	21,60
32	11507	81
33	11935	3,294,103,28
34	11992	57
35	12141	116,33
36	12648	507
37	12977	254,75
38	13756	779
39	13945	189
40	14105	160
41	14387	282
42	14450	63
43	14470	16,4
44	14565	95
45	14875	21,289

Table A-9: Big-Bag-A for pB85766 using MaeI and HinfI

No	Fragments	Big Bag B for pB85766
1	11	2,9
2	751	164,48,423,105
3	1586	162,276,213,184
4	1672	86
5	1856	56,118,10
6	1983	50,77
7	2041	58
8	2155	114
9	2198	43
10	2388	190
11	2438	44,6
12	2546	108
13	2641	78,17
14	2741	100
15	3162	421
16	4138	65,262,649
17	4309	171
18	5462	521,312,320
19	5503	41
20	5969	466
21	6072	103
22	6508	319,117
23	6743	235
24	8126	660,108,86,224,305
25	8142	16
26	8238	96
27	8456	218
28	8982	526
29	9092	55,55
30	9199	107
31	9618	174,36,209
32	11366	196,261,510,322,288,150,21
33	11510	60,81,3
34	11804	294
35	11907	103
36	12108	28,57,116
37	12902	33,507,254
38	14466	75,779,189,160,282,63,16
39	14586	4,95,21
40	14875	289

Table A-10: Big-Bag-B for pB85766 using MaeI and HinfI

B. Output Tables

BA. Output of Fragment Assembly of pB85766 when difference between number of two big bags is high

Node No	CurrBag	S	Options	SelBag	Choices	Cflag
1	A1	{77 297 153}	{B3 B2 B1}	{B3}	{B2 B1}	1
2	B3	{75 70 297 153}	{B3 B2 B1}	{A10}	{B2 B1}	1
3	A10	{281 75 70 297 153}	{B10 B2 B1}	{B10}	{B2 B1}	1
4	B10	{190 75 70 143 153 297}	{A11 A2 A3 A9}	{A11}	{A2 A3 A9}	1
5	A11	{75 70 64 143 153 297}	{A11 A2 A3 A9}	{A12}	{A2 A3 A9}	1
6	A12	{75 70 64 386 143 153 10 297}	{B11 B12 B1 B11 B2}	{B11}	{B12 B1 B2}	1
7	B11	{75 70 386 143 153 297}	{B11 B12 B1 B11 B2}	{A13}	{B12 B1 B2}	1
8	A13	{70 386 143 297 636 75 153 15 392 256 19}	{B12 B2 B15 B1 B16 B14 B12 B13}	{B12}	{B2 B15 B1 B16 B14 B13}	1
9	B12	{70 143 297 636 75 153 15 392 19}	{B2 B15 B1 B16 B14 B13}	{B2}	{B15 B1 B16 B14 B13}	1
10	B2	{70 143 636 75 153 15 392 19}	{B15 B1 B16 B14 B13}	{B15}	{B1 B16 B14 B13}	1
11	B15	{70 143 75 153 15 392 19}	{B1 B16 B14 B13}	{B1}	{B16 B14 B13}	1
12	B1	{70 143 75 15 392 19}	{B1 B16 B14 B13}	{A14}	{B16 B14 B13}	1
13	A14	{70 385 143 147 75 15 392 19}	{B17 B16 B28 B16 B14 B13}	{B17}	{B16 B28 B16 B14 B13}	1
14	B17	{11 70 143 147 75 15 392 19}	{B17 B16 B28 B16 B14 B13}	{A15}	{B16 B28 B16 B14 B13}	1
15	A15	{70 287 143 96 147 75 15 392 19}	{B18 B19 B16 B28 B16 B14 B13}	{B18}	{B19 B16 B28 B16 B14 B13}	1
16	B18	{70 143 96 147 75 15 392 19}	{B19 B16 B28 B16 B14 B13}	{B19}	{B16 B28 B16 B14 B13}	1

17	B19	{152 70 143 344 147 75 15 392 19}	{A16 A8 A3 A9 A17 A25 A2}	{A16}	{A8 A3 A9 A17 A25 A2}	1
18	A16	{70 143 344 147 75 15 392 19}	{A16 A8 A3 A9 A17 A25 A2}	{A17}	{A8 A3 A9 A17 A25 A2}	1
19	A17	{53 70 143 30 147 75 390 15 392 317 45 19}	{B22 B33 B9 B23 B16 B28 B21 B16 B14 B24 B20 B13}	{B22}	{B33 B9 B23 B16 B28 B21 B16 B14 B24 B20 B13}	1
20	B22	{70 143 30 147 75 390 15 392 317 45 19}	{B23 B16 B28 B21 B16 B14 B24 B20 B13}	{B23}	{B16 B28 B21 B16 B14 B24 B20 B13}	1
21	B23	{70 143 147 75 390 15 392 317 45 19}	{B16 B28 B21 B16 B14 B24 B20 B13}	{B16}	{B28 B21 B14 B24 B20 B13}	1
22	B16	{70 143 75 390 392 317 45 19}	{B21 B14 B24 B20 B13}	{B21}	{B14 B24 B20 B13}	1
23	B21	{70 143 75 392 317 45 19}	{B14 B24 B20 B13}	{B14}	{B24 B20 B13}	1
24	B14	{70 143 75 317 45 19}	{B14 B24 B20 B13}	{A18}	{B24 B20 B13}	1
25	A18	{70 143 18 75 41 317 45 19}	{B25 B24 B24 B20 B13}	{B25}	{B24 B24 B20 B13}	1
26	B25	{70 1063 182 75 143 41 317 45 19}	{A3 A20 A19 A2 A9}	{A3}	{A20 A19 A2 A9}	1
27	A3	{1063 182 75 215 143 41 317 45 19}	{B4 B24 B24 B20 B13}	{B4}	{B24 B24 B20 B13}	1
28	B4	{384 1063 125 182 75 418 143 41 317 45 19}	{A4 A20 A5 A19 A2 A6 A9}	{A4}	{A20 A5 A19 A2 A6 A9}	1
29	A4	{1063 125 182 75 418 143 41 317 45 19}	{A20 A5 A19 A2 A6 A9}	{A20}	{A5 A19 A2 A6 A9}	1
30	A20	{125 182 75 418 143 13 840 41 317 45 19}	{B27 B26 B24 B24 B20 B13}	{B27}	{B26 B24 B24 B20 B13}	1
31	B27	{68 125 44 182 75 418 24 143 174 840 41 317 45 19}	{A21 A7 A5 A23 A19 A2 A6 A24 A36 A45 A9 A22}	{A21}	{A7 A5 A23 A19 A2 A6 A24 A36 A45 A9 A22}	1
32	A21	{125 44 182 75 418 24 143 174 840 41 317 45 19}	{A5 A23 A19 A2 A6 A24 A36 A45 A9 A22}	{A5}	{A23 A19 A2 A6 A24 A36 A45 A9 A22}	1

33	A5	{44 182 75 418 24 143 174 840 41 317 45 19}	{A23 A19 A2 A6 A24 A36 A45 A9 A22}	{A23}	{A19 A2 A6 A24 A36 A45 A9 A22}	1
34	A23	{182 75 418 24 143 174 840 41 317 45 19}	{A19 A2 A6 A24 A36 A45 A9 A22}	{A19}	{A2 A6 A24 A36 A45 A9 A22}	1
35	A19	{75 418 24 143 174 840 41 317 45 19}	{A19 A2 A6 A24 A36 A45 A9 A22}	{A2}	{A2 A6 A24 A36 A45 A9 A22}	1
36	A2	{418 24 143 174 840 41 317 45 19}	{B32 B33 B26 B24 B24 B20 B13}	{B32}	{B33 B26 B24 B24 B20 B13}	1
37	B32	{453 83 418 131 130 239 78 143 174 235 840 41 47 38 317 45 507 19}	{A44 A38 A6 A43 A34 A30 A41 A9 A22 A35 A42 A31 A33}	{A44}	{A38 A6 A43 A34 A30 A41 A9 A22 A35 A42 A31 A33}	1
38	A44	{83 418 131 130 239 78 143 360 174 235 840 41 47 38 317 45 507 19}	{A38 A6 A43 A34 A30 A41 A9 A22 A35 A42 A31 A33}	{A38}	{A6 A43 A34 A30 A41 A9 A22 A35 A42 A31 A33}	1
39	A38	{418 131 130 239 78 143 360 174 235 840 41 47 38 317 45 507 19}	{A6 A43 A34 A30 A41 A9 A22 A35 A42 A31 A33}	{A6}	{A43 A34 A30 A41 A9 A22 A35 A42 A31 A33}	1
40	A6	{131 130 330 239 78 143 360 174 235 840 41 47 38 317 45 507 19}	{A43 A34 A30 A41 A9 A22 A35 A42 A31 A33}	{A43}	{A34 A30 A41 A9 A22 A35 A42 A31 A33}	1
41	A43	{130 330 239 78 143 360 174 235 840 41 47 38 317 45 507 19}	{A34 A30 A41 A9 A22 A35 A42 A31 A33}	{A34}	{A30 A41 A9 A22 A35 A42 A31 A33}	1
42	A34	{330 239 78 143 360 174 235 840 41 47 38 317 45 507 19}	{A30 A41 A9 A22 A35 A42 A31 A33}	{A30}	{A41 A9 A22 A35 A42 A31 A33}	1
43	A30	{495 162 330 78 143 360 174 235 840 41 47 38 317 45 507 19}	{A41 A9 A22 A35 A42 A31 A33}	{A41}	{A9 A22 A35 A42 A31 A33}	1
44	A41	{495 162 330 143 360 174 235 840 41 47 38 317 45 507 19}	{A9 A22 A35 A42 A31 A33}	{A9}	{A22 A35 A42 A31 A33}	1
45	A9	{53 495 162 89 330 360 174 235 840 41 47 38 317 45 507 19}	{A9 A22 A35 A42 A31 A33}	{A22}	{A22 A35 A42 A31 A33}	1

46	A22	{53 495 162 89 330 360 235 840 41 47 38 317 45 507 19}	{B33 B9 B30 B31 B8 B5 B33 B26 B24 B24 B20 B13}	{B33}	{B9 B30 B31 B8 B5 B26 B24 B24 B20 B13}	1
47	B33	{495 162 59 89 330 235 840 41 47 38 317 45 507 19}	{A51 A35 A42 A31 A33}	{A51}	{A35 A42 A31 A33}	1
48	A51	{495 162 89 330 235 840 41 47 38 317 45 507 19}	{A35 A42 A31 A33}	{A35}	{A42 A31 A33}	1
49	A35	{495 162 89 330 840 41 47 38 317 45 507 19}	{A42 A31 A33}	{A42}	{A31 A33}	1
50	A42	{495 162 89 330 840 41 38 317 45 507 19}	{A31 A33}	{A31}	{A33}	1
51	A31	{495 162 89 330 840 41 317 45 507 19}	{A33}	{A33}	{}	0
BT=>49	A35	{495 162 89 330 840 41 47 38 317 45 507 19}	{A31 A33}	{A31}	{A33}	1
49	A35	{495 162 89 330 840 41 47 38 317 45 507 19}	{A31 A33}	{A31}	{A33}	1
50	A31	{495 162 89 330 840 41 47 317 45 507 19}	{A42 A33}	{A42}	{A33}	1
51	A42	{495 162 89 330 840 41 317 45 507 19}	{A33}	{A33}	{}	0
BT=>49	A35	{495 162 89 330 840 41 47 38 317 45 507 19}	{A33}	{A33}	{}	0
49	A35	{495 162 89 330 840 41 47 38 317 45 507 19}	{A33}	{A33}	{}	0
50	A33	{495 162 89 330 840 41 47 38 317 45 19}	{A42 A31}	{A42}	{A31}	1
51	A42	{495 162 89 330 840 41 38 317 45 19}	{A31}	{A31}	{}	0
BT=>48	A51	{495 162 89 330 235 840 41 47 38 317 45 507 19}	{A31 A33 A42}	{A31}	{A33 A42}	1

48	A51	{495 162 89 330 235 840 41 47 38 317 45 507 19}	{A31 A33 A42}	{A31}	{A33 A42}	1
49	A31	{495 162 89 330 235 840 41 47 317 45 507 19}	{A35 A42 A33}	{A35}	{A42 A33}	1
50	A35	{495 162 89 330 840 41 47 317 45 507 19}	{A42 A33}	{A42}	{A33}	1
51	A42	{495 162 89 330 840 41 317 45 507 19}	{A33}	{A33}	{}	0
BT=>49	A31	{495 162 89 330 235 840 41 47 317 45 507 19}	{A33 A42}	{A33}	{A42}	1
49	A31	{495 162 89 330 235 840 41 47 317 45 507 19}	{A33 A42}	{A33}	{A42}	1
50	A33	{495 162 89 330 235 840 41 47 317 45 19}	{A35 A42}	{A35}	{A42}	1
51	A35	{495 162 89 330 840 41 47 317 45 19}	{A42}	{A42}	{}	0
BT=>49	A31	{495 162 89 330 235 840 41 47 317 45 507 19}	{A42}	{A42}	{}	0
49	A31	{495 162 89 330 235 840 41 47 317 45 507 19}	{A42}	{A42}	{}	0
50	A42	{495 162 89 330 235 840 41 317 45 507 19}	{A35 A33}	{A35}	{A33}	1
51	A35	{495 162 89 330 840 41 317 45 507 19}	{A33}	{A33}	{}	0
BT=>48	A51	{495 162 89 330 235 840 41 47 38 317 45 507 19}	{A33 A42}	{A33}	{A42}	1
48	A51	{495 162 89 330 235 840 41 47 38 317 45 507 19}	{A33 A42}	{A33}	{A42}	1
49	A33	{495 162 89 330 235 840 41 47 38 317 45 19}	{A35 A42 A31}	{A35}	{A42 A31}	1
50	A35	{495 162 89 330 840 41 47 38 317 45 19}	{A42 A31}	{A42}	{A31}	1

51	A42	{495 162 89 330 840 41 38 317 45 19}	{A31}	{A31}	{}	0
BT=>49	A33	{495 162 89 330 235 840 41 47 38 317 45 19}	{A31 A42}	{A31}	{A42}	1
49	A33	{495 162 89 330 235 840 41 47 38 317 45 19}	{A31 A42}	{A31}	{A42}	1
50	A31	{495 162 89 330 235 840 41 47 317 45 19}	{A35 A42}	{A35}	{A42}	1
51	A35	{495 162 89 330 840 41 47 317 45 19}	{A42}	{A42}	{}	0
BT=>49	A33	{495 162 89 330 235 840 41 47 38 317 45 19}	{A42}	{A42}	{}	0
49	A33	{495 162 89 330 235 840 41 47 38 317 45 19}	{A42}	{A42}	{}	0
50	A42	{495 162 89 330 235 840 41 38 317 45 19}	{A35 A31}	{A35}	{A31}	1
51	A35	{495 162 89 330 840 41 38 317 45 19}	{A31}	{A31}	{}	0
BT=>48	A51	{495 162 89 330 235 840 41 47 38 317 45 507 19}	{A42}	{A42}	{}	0
48	A51	{495 162 89 330 235 840 41 47 38 317 45 507 19}	{A42}	{A42}	{}	0
49	A42	{495 162 89 330 235 840 41 38 317 45 507 19}	{A35 A31 A33}	{A35}	{A31 A33}	1
50	A35	{495 162 89 330 840 41 38 317 45 507 19}	{A31 A33}	{A31}	{A33}	1
51	A31	{495 162 89 330 840 41 317 45 507 19}	{A33}	{A33}	{}	0
BT=>49	A42	{495 162 89 330 235 840 41 38 317 45 507 19}	{A31 A33}	{A31}	{A33}	1
49	A42	{495 162 89 330 235 840 41 38 317 45 507 19}	{A31 A33}	{A31}	{A33}	1

50	A31	{495 162 89 330 235 840 41 317 45 507 19}	{A35 A33}	{A35}	{A33}	1
51	A35	{495 162 89 330 840 41 317 45 507 19}	{A33}	{A33}	{}	0
BT=>49	A42	{495 162 89 330 235 840 41 38 317 45 507 19}	{A33}	{A33}	{}	0
49	A42	{495 162 89 330 235 840 41 38 317 45 507 19}	{A33}	{A33}	{}	0
50	A33	{495 162 89 330 235 840 41 38 317 45 19}	{A35 A31}	{A35}	{A31}	1
51	A35	{495 162 89 330 840 41 38 317 45 19}	{A31}	{A31}	{}	0
BT=>47	B33	{495 162 59 89 330 235 840 41 47 38 317 45 507 19}	{A31 A33 A35 A42}	{A31}	{A33 A35 A42}	1
47	B33	{495 162 59 89 330 235 840 41 47 38 317 45 507 19}	{A31 A33 A35 A42}	{A31}	{A33 A35 A42}	1
48	A31	{495 162 59 89 330 235 840 41 47 317 45 507 19}	{A51 A35 A42 A33}	{A51}	{A35 A42 A33}	1
49	A51	{495 162 89 330 235 840 41 47 317 45 507 19}	{A35 A42 A33}	{A35}	{A42 A33}	1
50	A35	{495 162 89 330 840 41 47 317 45 507 19}	{A42 A33}	{A42}	{A33}	1
51	A42	{495 162 89 330 840 41 317 45 507 19}	{A33}	{A33}	{}	0
BT=>49	A51	{495 162 89 330 235 840 41 47 317 45 507 19}	{A33 A42}	{A33}	{A42}	1
49	A51	{495 162 89 330 235 840 41 47 317 45 507 19}	{A33 A42}	{A33}	{A42}	1
50	A33	{495 162 89 330 235 840 41 47 317 45 19}	{A35 A42}	{A35}	{A42}	1
51	A35	{495 162 89 330 840 41 47 317 45 19}	{A42}	{A42}	{}	0

BT=>49	A51	{495 162 89 330 235 840 41 47 317 45 507 19}	{A42}	{A42}	{}	0
49	A51	{495 162 89 330 235 840 41 47 317 45 507 19}	{A42}	{A42}	{}	0
50	A42	{495 162 89 330 235 840 41 317 45 507 19}	{A35 A33}	{A35}	{A33}	1
51	A35	{495 162 89 330 840 41 317 45 507 19}	{A33}	{A33}	{}	0
BT=>48	A31	{495 162 59 89 330 235 840 41 47 317 45 507 19}	{A33 A35 A42}	{A33}	{A35 A42}	1
48	A31	{495 162 59 89 330 235 840 41 47 317 45 507 19}	{A33 A35 A42}	{A33}	{A35 A42}	1
49	A33	{495 162 59 89 330 235 840 41 47 317 45 19}	{A51 A35 A42}	{A51}	{A35 A42}	1
50	A51	{495 162 89 330 235 840 41 47 317 45 19}	{A35 A42}	{A35}	{A42}	1
51	A35	{495 162 89 330 840 41 47 317 45 19}	{A42}	{A42}	{}	0
BT=>49	A33	{495 162 59 89 330 235 840 41 47 317 45 19}	{A35 A42}	{A35}	{A42}	1
49	A33	{495 162 59 89 330 235 840 41 47 317 45 19}	{A35 A42}	{A35}	{A42}	1
50	A35	{495 162 59 89 330 840 41 47 317 45 19}	{A51 A42}	{A51}	{A42}	1
51	A51	{495 162 89 330 840 41 47 317 45 19}	{A42}	{A42}	{}	0
BT=>49	A33	{495 162 59 89 330 235 840 41 47 317 45 19}	{A42}	{A42}	{}	0
49	A33	{495 162 59 89 330 235 840 41 47 317 45 19}	{A42}	{A42}	{}	0
50	A42	{495 162 59 89 330 235 840 41 317 45 19}	{A51 A35}	{A51}	{A35}	1

51	A51	{495 162 89 330 235 840 41 317 45 19}	{A35}	{A35}	{}	0
BT=>48	A31	{495 162 59 89 330 235 840 41 47 317 45 507 19}	{A35 A42}	{A35}	{A42}	1
48	A31	{495 162 59 89 330 235 840 41 47 317 45 507 19}	{A35 A42}	{A35}	{A42}	1
49	A35	{495 162 59 89 330 840 41 47 317 45 507 19}	{A51 A42 A33}	{A51}	{A42 A33}	1
50	A51	{495 162 89 330 840 41 47 317 45 507 19}	{A42 A33}	{A42}	{A33}	1
51	A42	{495 162 89 330 840 41 317 45 507 19}	{A33}	{A33}	{}	0
BT=>49	A35	{495 162 59 89 330 840 41 47 317 45 507 19}	{A33 A42}	{A33}	{A42}	1
49	A35	{495 162 59 89 330 840 41 47 317 45 507 19}	{A33 A42}	{A33}	{A42}	1
50	A33	{495 162 59 89 330 840 41 47 317 45 19}	{A51 A42}	{A51}	{A42}	1
51	A51	{495 162 89 330 840 41 47 317 45 19}	{A42}	{A42}	{}	0
BT=>49	A35	{495 162 59 89 330 840 41 47 317 45 507 19}	{A42}	{A42}	{}	0
49	A35	{495 162 59 89 330 840 41 47 317 45 507 19}	{A42}	{A42}	{}	0
50	A42	{495 162 59 89 330 840 41 317 45 507 19}	{A51 A33}	{A51}	{A33}	1
51	A51	{495 162 89 330 840 41 317 45 507 19}	{A33}	{A33}	{}	0
BT=>48	A31	{495 162 59 89 330 235 840 41 47 317 45 507 19}	{A42}	{A42}	{}	0
48	A31	{495 162 59 89 330 235 840 41 47 317 45 507 19}	{A42}	{A42}	{}	0

49	A42	{495 162 59 89 330 235 840 41 317 45 507 19}	{A51 A35 A33}	{A51}	{A35 A33}	1
50	A51	{495 162 89 330 235 840 41 317 45 507 19}	{A35 A33}	{A35}	{A33}	1
51	A35	{495 162 89 330 840 41 317 45 507 19}	{A33}	{A33}	{}	0
BT=>49	A42	{495 162 59 89 330 235 840 41 317 45 507 19}	{A33 A35}	{A33}	{A35}	1
49	A42	{495 162 59 89 330 235 840 41 317 45 507 19}	{A33 A35}	{A33}	{A35}	1
50	A33	{495 162 59 89 330 235 840 41 317 45 19}	{A51 A35}	{A51}	{A35}	1
51	A51	{495 162 89 330 235 840 41 317 45 19}	{A35}	{A35}	{}	0
BT=>49	A42	{495 162 59 89 330 235 840 41 317 45 507 19}	{A35}	{A35}	{}	0
49	A42	{495 162 59 89 330 235 840 41 317 45 507 19}	{A35}	{A35}	{}	0
50	A35	{495 162 59 89 330 840 41 317 45 507 19}	{A51 A33}	{A51}	{A33}	1
51	A51	{495 162 89 330 840 41 317 45 507 19}	{A33}	{A33}	{}	0
BT=>47	B33	{495 162 59 89 330 235 840 41 47 38 317 45 507 19}	{A33 A35 A42}	{A33}	{A35 A42}	1
47	B33	{495 162 59 89 330 235 840 41 47 38 317 45 507 19}	{A33 A35 A42}	{A33}	{A35 A42}	1
48	A33	{495 162 59 89 330 235 840 41 47 38 317 45 19}	{A51 A35 A42 A31}	{A51}	{A35 A42 A31}	1
49	A51	{495 162 89 330 235 840 41 47 38 317 45 19}	{A35 A42 A31}	{A35}	{A42 A31}	1
50	A35	{495 162 89 330 840 41 47 38 317 45 19}	{A42 A31}	{A42}	{A31}	1

51	A42	{495 162 89 330 840 41 38 317 45 19}	{A31}	{A31}	{}	0
BT=>49	A51	{495 162 89 330 235 840 41 47 38 317 45 19}	{A31 A42}	{A31}	{A42}	1
49	A51	{495 162 89 330 235 840 41 47 38 317 45 19}	{A31 A42}	{A31}	{A42}	1
50	A31	{495 162 89 330 235 840 41 47 317 45 19}	{A35 A42}	{A35}	{A42}	1
51	A35	{495 162 89 330 840 41 47 317 45 19}	{A42}	{A42}	{}	0
BT=>49	A51	{495 162 89 330 235 840 41 47 38 317 45 19}	{A42}	{A42}	{}	0
49	A51	{495 162 89 330 235 840 41 47 38 317 45 19}	{A42}	{A42}	{}	0
50	A42	{495 162 89 330 235 840 41 38 317 45 19}	{A35 A31}	{A35}	{A31}	1
51	A35	{495 162 89 330 840 41 38 317 45 19}	{A31}	{A31}	{}	0
BT=>48	A33	{495 162 59 89 330 235 840 41 47 38 317 45 19}	{A31 A35 A42}	{A31}	{A35 A42}	1
48	A33	{495 162 59 89 330 235 840 41 47 38 317 45 19}	{A31 A35 A42}	{A31}	{A35 A42}	1
49	A31	{495 162 59 89 330 235 840 41 47 317 45 19}	{A51 A35 A42}	{A51}	{A35 A42}	1
50	A51	{495 162 89 330 235 840 41 47 317 45 19}	{A35 A42}	{A35}	{A42}	1
51	A35	{495 162 89 330 840 41 47 317 45 19}	{A42}	{A42}	{}	0
BT=>49	A31	{495 162 59 89 330 235 840 41 47 317 45 19}	{A35 A42}	{A35}	{A42}	1
49	A31	{495 162 59 89 330 235 840 41 47 317 45 19}	{A35 A42}	{A35}	{A42}	1

50	A35	{495 162 59 89 330 840 41 47 317 45 19}	{A51 A42}	{A51}	{A42}	1
51	A51	{495 162 89 330 840 41 47 317 45 19}	{A42}	{A42}	{}	0
BT=>49	A31	{495 162 59 89 330 235 840 41 47 317 45 19}	{A42}	{A42}	{}	0
49	A31	{495 162 59 89 330 235 840 41 47 317 45 19}	{A42}	{A42}	{}	0
50	A42	{495 162 59 89 330 235 840 41 317 45 19}	{A51 A35}	{A51}	{A35}	1
51	A51	{495 162 89 330 235 840 41 317 45 19}	{A35}	{A35}	{}	0
BT=>48	A33	{495 162 59 89 330 235 840 41 47 38 317 45 19}	{A35 A42}	{A35}	{A42}	1
48	A33	{495 162 59 89 330 235 840 41 47 38 317 45 19}	{A35 A42}	{A35}	{A42}	1
49	A35	{495 162 59 89 330 840 41 47 38 317 45 19}	{A51 A42 A31}	{A51}	{A42 A31}	1
50	A51	{495 162 89 330 840 41 47 38 317 45 19}	{A42 A31}	{A42}	{A31}	1
51	A42	{495 162 89 330 840 41 38 317 45 19}	{A31}	{A31}	{}	0
BT=>49	A35	{495 162 59 89 330 840 41 47 38 317 45 19}	{A31 A42}	{A31}	{A42}	1
49	A35	{495 162 59 89 330 840 41 47 38 317 45 19}	{A31 A42}	{A31}	{A42}	1
50	A31	{495 162 59 89 330 840 41 47 317 45 19}	{A51 A42}	{A51}	{A42}	1
51	A51	{495 162 89 330 840 41 47 317 45 19}	{A42}	{A42}	{}	0
BT=>49	A35	{495 162 59 89 330 840 41 47 38 317 45 19}	{A42}	{A42}	{}	0

49	A35	{495 162 59 89 330 840 41 47 38 317 45 19}	{A42}	{A42}	{}	0
50	A42	{495 162 59 89 330 840 41 38 317 45 19}	{A51 A31}	{A51}	{A31}	1
51	A51	{495 162 89 330 840 41 38 317 45 19}	{A31}	{A31}	{}	0
BT=>48	A33	{495 162 59 89 330 235 840 41 47 38 317 45 19}	{A42}	{A42}	{}	0
48	A33	{495 162 59 89 330 235 840 41 47 38 317 45 19}	{A42}	{A42}	{}	0
49	A42	{495 162 59 89 330 235 840 41 38 317 45 19}	{A51 A35 A31}	{A51}	{A35 A31}	1
50	A51	{495 162 89 330 235 840 41 38 317 45 19}	{A35 A31}	{A35}	{A31}	1
51	A35	{495 162 89 330 840 41 38 317 45 19}	{A31}	{A31}	{}	0
BT=>49	A42	{495 162 59 89 330 235 840 41 38 317 45 19}	{A31 A35}	{A31}	{A35}	1
49	A42	{495 162 59 89 330 235 840 41 38 317 45 19}	{A31 A35}	{A31}	{A35}	1
50	A31	{495 162 59 89 330 235 840 41 317 45 19}	{A51 A35}	{A51}	{A35}	1
51	A51	{495 162 89 330 235 840 41 317 45 19}	{A35}	{A35}	{}	0
BT=>49	A42	{495 162 59 89 330 235 840 41 38 317 45 19}	{A35}	{A35}	{}	0
49	A42	{495 162 59 89 330 235 840 41 38 317 45 19}	{A35}	{A35}	{}	0
50	A35	{495 162 59 89 330 840 41 38 317 45 19}	{A51 A31}	{A51}	{A31}	1
51	A51	{495 162 89 330 840 41 38 317 45 19}	{A31}	{A31}	{}	0

BT=>47	B33	{495 162 59 89 330 235 840 41 47 38 317 45 507 19}	{A35 A42}	{A35}	{A42}	1
47	B33	{495 162 59 89 330 235 840 41 47 38 317 45 507 19}	{A35 A42}	{A35}	{A42}	1
48	A35	{495 162 59 89 330 840 41 47 38 317 45 507 19}	{A51 A42 A31 A33}	{A51}	{A42 A31 A33}	1
49	A51	{495 162 89 330 840 41 47 38 317 45 507 19}	{A42 A31 A33}	{A42}	{A31 A33}	1
50	A42	{495 162 89 330 840 41 38 317 45 507 19}	{A31 A33}	{A31}	{A33}	1
51	A31	{495 162 89 330 840 41 317 45 507 19}	{A33}	{A33}	{}	0
BT=>49	A51	{495 162 89 330 840 41 47 38 317 45 507 19}	{A31 A33}	{A31}	{A33}	1
49	A51	{495 162 89 330 840 41 47 38 317 45 507 19}	{A31 A33}	{A31}	{A33}	1
50	A31	{495 162 89 330 840 41 47 317 45 507 19}	{A42 A33}	{A42}	{A33}	1
51	A42	{495 162 89 330 840 41 317 45 507 19}	{A33}	{A33}	{}	0
BT=>49	A51	{495 162 89 330 840 41 47 38 317 45 507 19}	{A33}	{A33}	{}	0
49	A51	{495 162 89 330 840 41 47 38 317 45 507 19}	{A33}	{A33}	{}	0
50	A33	{495 162 89 330 840 41 47 38 317 45 19}	{A42 A31}	{A42}	{A31}	1
51	A42	{495 162 89 330 840 41 38 317 45 19}	{A31}	{A31}	{}	0
BT=>48	A35	{495 162 59 89 330 840 41 47 38 317 45 507 19}	{A31 A33 A42}	{A31}	{A33 A42}	1
48	A35	{495 162 59 89 330 840 41 47 38 317 45 507 19}	{A31 A33 A42}	{A31}	{A33 A42}	1

49	A31	{495 162 59 89 330 840 41 47 317 45 507 19}	{A51 A42 A33}	{A51}	{A42 A33}	1
50	A51	{495 162 89 330 840 41 47 317 45 507 19}	{A42 A33}	{A42}	{A33}	1
51	A42	{495 162 89 330 840 41 317 45 507 19}	{A33}	{A33}	{}	0
BT=>49	A31	{495 162 59 89 330 840 41 47 317 45 507 19}	{A33 A42}	{A33}	{A42}	1
49	A31	{495 162 59 89 330 840 41 47 317 45 507 19}	{A33 A42}	{A33}	{A42}	1
50	A33	{495 162 59 89 330 840 41 47 317 45 19}	{A51 A42}	{A51}	{A42}	1
51	A51	{495 162 89 330 840 41 47 317 45 19}	{A42}	{A42}	{}	0
BT=>49	A31	{495 162 59 89 330 840 41 47 317 45 507 19}	{A42}	{A42}	{}	0
49	A31	{495 162 59 89 330 840 41 47 317 45 507 19}	{A42}	{A42}	{}	0
50	A42	{495 162 59 89 330 840 41 317 45 507 19}	{A51 A33}	{A51}	{A33}	1
51	A51	{495 162 89 330 840 41 317 45 507 19}	{A33}	{A33}	{}	0
BT=>48	A35	{495 162 59 89 330 840 41 47 38 317 45 507 19}	{A33 A42}	{A33}	{A42}	1
48	A35	{495 162 59 89 330 840 41 47 38 317 45 507 19}	{A33 A42}	{A33}	{A42}	1
49	A33	{495 162 59 89 330 840 41 47 38 317 45 19}	{A51 A42 A31}	{A51}	{A42 A31}	1
50	A51	{495 162 89 330 840 41 47 38 317 45 19}	{A42 A31}	{A42}	{A31}	1
51	A42	{495 162 89 330 840 41 38 317 45 19}	{A31}	{A31}	{}	0

BT=>49	A33	{495 162 59 89 330 840 41 47 38 317 45 19}	{A31 A42}	{A31}	{A42}	1
49	A33	{495 162 59 89 330 840 41 47 38 317 45 19}	{A31 A42}	{A31}	{A42}	1
50	A31	{495 162 59 89 330 840 41 47 317 45 19}	{A51 A42}	{A51}	{A42}	1
51	A51	{495 162 89 330 840 41 47 317 45 19}	{A42}	{A42}	{}	0
BT=>49	A33	{495 162 59 89 330 840 41 47 38 317 45 19}	{A42}	{A42}	{}	0
49	A33	{495 162 59 89 330 840 41 47 38 317 45 19}	{A42}	{A42}	{}	0
50	A42	{495 162 59 89 330 840 41 38 317 45 19}	{A51 A31}	{A51}	{A31}	1
51	A51	{495 162 89 330 840 41 38 317 45 19}	{A31}	{A31}	{}	0
BT=>48	A35	{495 162 59 89 330 840 41 47 38 317 45 507 19}	{A42}	{A42}	{}	0
48	A35	{495 162 59 89 330 840 41 47 38 317 45 507 19}	{A42}	{A42}	{}	0
49	A42	{495 162 59 89 330 840 41 38 317 45 507 19}	{A51 A31 A33}	{A51}	{A31 A33}	1
50	A51	{495 162 89 330 840 41 38 317 45 507 19}	{A31 A33}	{A31}	{A33}	1
51	A31	{495 162 89 330 840 41 317 45 507 19}	{A33}	{A33}	{}	0
BT=>49	A42	{495 162 59 89 330 840 41 38 317 45 507 19}	{A31 A33}	{A31}	{A33}	1
49	A42	{495 162 59 89 330 840 41 38 317 45 507 19}	{A31 A33}	{A31}	{A33}	1
50	A31	{495 162 59 89 330 840 41 317 45 507 19}	{A51 A33}	{A51}	{A33}	1

51	A51	{495 162 89 330 840 41 317 45 507 19}	{A33}	{A33}	{}	0
BT=>49	A42	{495 162 59 89 330 840 41 38 317 45 507 19}	{A33}	{A33}	{}	0
49	A42	{495 162 59 89 330 840 41 38 317 45 507 19}	{A33}	{A33}	{}	0
50	A33	{495 162 59 89 330 840 41 38 317 45 19}	{A51 A31}	{A51}	{A31}	1
51	A51	{495 162 89 330 840 41 38 317 45 19}	{A31}	{A31}	{}	0
BT=>47	B33	{495 162 59 89 330 235 840 41 47 38 317 45 507 19}	{A42}	{A42}	{}	0
47	B33	{495 162 59 89 330 235 840 41 47 38 317 45 507 19}	{A42}	{A42}	{}	0
48	A42	{495 162 59 89 330 235 840 41 38 317 45 507 19}	{A51 A35 A31 A33}	{A51}	{A35 A31 A33}	1
49	A51	{495 162 89 330 235 840 41 38 317 45 507 19}	{A35 A31 A33}	{A35}	{A31 A33}	1
50	A35	{495 162 89 330 840 41 38 317 45 507 19}	{A31 A33}	{A31}	{A33}	1
51	A31	{495 162 89 330 840 41 317 45 507 19}	{A33}	{A33}	{}	0
BT=>49	A51	{495 162 89 330 235 840 41 38 317 45 507 19}	{A31 A33}	{A31}	{A33}	1
49	A51	{495 162 89 330 235 840 41 38 317 45 507 19}	{A31 A33}	{A31}	{A33}	1
50	A31	{495 162 89 330 235 840 41 317 45 507 19}	{A35 A33}	{A35}	{A33}	1
51	A35	{495 162 89 330 840 41 317 45 507 19}	{A33}	{A33}	{}	0
BT=>49	A51	{495 162 89 330 235 840 41 38 317 45 507 19}	{A33}	{A33}	{}	0

49	A51	{495 162 89 330 235 840 41 38 317 45 507 19}	{A33}	{A33}	{}	0
50	A33	{495 162 89 330 235 840 41 38 317 45 19}	{A35 A31}	{A35}	{A31}	1
51	A35	{495 162 89 330 840 41 38 317 45 19}	{A31}	{A31}	{}	0
BT=>48	A42	{495 162 59 89 330 235 840 41 38 317 45 507 19}	{A31 A33 A35}	{A31}	{A33 A35}	1
48	A42	{495 162 59 89 330 235 840 41 38 317 45 507 19}	{A31 A33 A35}	{A31}	{A33 A35}	1
49	A31	{495 162 59 89 330 235 840 41 317 45 507 19}	{A51 A35 A33}	{A51}	{A35 A33}	1
50	A51	{495 162 89 330 235 840 41 317 45 507 19}	{A35 A33}	{A35}	{A33}	1
51	A35	{495 162 89 330 840 41 317 45 507 19}	{A33}	{A33}	{}	0
BT=>49	A31	{495 162 59 89 330 235 840 41 317 45 507 19}	{A33 A35}	{A33}	{A35}	1
49	A31	{495 162 59 89 330 235 840 41 317 45 507 19}	{A33 A35}	{A33}	{A35}	1
50	A33	{495 162 59 89 330 235 840 41 317 45 19}	{A51 A35}	{A51}	{A35}	1
51	A51	{495 162 89 330 235 840 41 317 45 19}	{A35}	{A35}	{}	0
BT=>49	A31	{495 162 59 89 330 235 840 41 317 45 507 19}	{A35}	{A35}	{}	0
49	A31	{495 162 59 89 330 235 840 41 317 45 507 19}	{A35}	{A35}	{}	0
50	A35	{495 162 59 89 330 840 41 317 45 507 19}	{A51 A33}	{A51}	{A33}	1
51	A51	{495 162 89 330 840 41 317 45 507 19}	{A33}	{A33}	{}	0

BT=>48	A42	{495 162 59 89 330 235 840 41 38 317 45 507 19}	{A33 A35}	{A33}	{A35}	1
48	A42	{495 162 59 89 330 235 840 41 38 317 45 507 19}	{A33 A35}	{A33}	{A35}	1
49	A33	{495 162 59 89 330 235 840 41 38 317 45 19}	{A51 A35 A31}	{A51}	{A35 A31}	1
50	A51	{495 162 89 330 235 840 41 38 317 45 19}	{A35 A31}	{A35}	{A31}	1
51	A35	{495 162 89 330 840 41 38 317 45 19}	{A31}	{A31}	{}	0
BT=>49	A33	{495 162 59 89 330 235 840 41 38 317 45 19}	{A31 A35}	{A31}	{A35}	1
49	A33	{495 162 59 89 330 235 840 41 38 317 45 19}	{A31 A35}	{A31}	{A35}	1
50	A31	{495 162 59 89 330 235 840 41 317 45 19}	{A51 A35}	{A51}	{A35}	1
51	A51	{495 162 89 330 235 840 41 317 45 19}	{A35}	{A35}	{}	0
BT=>49	A33	{495 162 59 89 330 235 840 41 38 317 45 19}	{A35}	{A35}	{}	0
49	A33	{495 162 59 89 330 235 840 41 38 317 45 19}	{A35}	{A35}	{}	0
50	A35	{495 162 59 89 330 840 41 38 317 45 19}	{A51 A31}	{A51}	{A31}	1
51	A51	{495 162 89 330 840 41 38 317 45 19}	{A31}	{A31}	{}	0
BT=>48	A42	{495 162 59 89 330 235 840 41 38 317 45 507 19}	{A35}	{A35}	{}	0
48	A42	{495 162 59 89 330 235 840 41 38 317 45 507 19}	{A35}	{A35}	{}	0
49	A35	{495 162 59 89 330 840 41 38 317 45 507 19}	{A51 A31 A33}	{A51}	{A31 A33}	1

50	A51	{495 162 89 330 840 41 38 317 45 507 19}	{A31 A33}	{A31}	{A33}	1
51	A31	{495 162 89 330 840 41 317 45 507 19}	{A33}	{A33}	{}	0
BT=>49	A35	{495 162 59 89 330 840 41 38 317 45 507 19}	{A31 A33}	{A31}	{A33}	1
49	A35	{495 162 59 89 330 840 41 38 317 45 507 19}	{A31 A33}	{A31}	{A33}	1
50	A31	{495 162 59 89 330 840 41 317 45 507 19}	{A51 A33}	{A51}	{A33}	1
51	A51	{495 162 89 330 840 41 317 45 507 19}	{A33}	{A33}	{}	0
BT=>49	A35	{495 162 59 89 330 840 41 38 317 45 507 19}	{A33}	{A33}	{}	0
49	A35	{495 162 59 89 330 840 41 38 317 45 507 19}	{A33}	{A33}	{}	0
50	A33	{495 162 59 89 330 840 41 38 317 45 19}	{A51 A31}	{A51}	{A31}	1
51	A51	{495 162 89 330 840 41 38 317 45 19}	{A31}	{A31}	{}	0
BT=>46	A22	{53 495 162 89 330 360 235 840 41 47 38 317 45 507 19}	{B13 B20 B24 B24 B26 B30 B31 B5 B8 B9}	{B13}	{B20 B24 B24 B26 B30 B31 B5 B8 B9}	1
46	A22	{53 495 162 89 330 360 235 840 41 47 38 317 45 507 19}	{B13 B20 B24 B24 B26 B30 B31 B5 B8 B9}	{B13}	{B20 B24 B24 B26 B30 B31 B5 B8 B9}	1
47	B13	{53 495 162 89 330 360 235 840 41 47 38 317 45 507}	{B13 B20 B24 B24 B26 B30 B31 B5 B8 B9}	{A24}	{B20 B24 B24 B26 B30 B31 B5 B8 B9}	1
48	A24	{53 24 495 162 89 330 360 235 85 840 41 47 38 317 45 507}	{B33 B9 B33 B30 B31 B8 B5 B33 B28 B26 B24 B24 B20}	{B33}	{B9 B30 B31 B8 B5 B28 B26 B24 B24 B20}	1
49	B33	{24 24 495 162 59 89 330 235 85 840 41 47 38 317 45 507}	{A36 A45 A36 A45 A51 A35 A42 A31 A33}	{A36}	{A45 A45 A51 A35 A42 A31 A33}	1

50	A36	{24 495 162 59 89 330 235 85 840 41 47 38 317 45 507}	{A45 A51 A35 A42 A31 A33}	{A45}	{A51 A35 A42 A31 A33}	1
51	A45	{495 162 59 89 330 235 85 840 41 47 38 317 45 507}	{A51 A35 A42 A31 A33}	{A51}	{A35 A42 A31 A33}	1
52	A51	{495 162 89 330 235 85 840 41 47 38 317 45 507}	{A35 A42 A31 A33}	{A35}	{A42 A31 A33}	1
53	A35	{495 162 89 330 85 840 41 47 38 317 45 507}	{A42 A31 A33}	{A42}	{A31 A33}	1
54	A42	{495 162 89 330 85 840 41 38 317 45 507}	{A31 A33}	{A31}	{A33}	1
55	A31	{495 162 89 330 85 840 41 317 45 507}	{A31 A33}	{A25}	{A33}	1
56	A25	{147 495 162 89 330 85 840 9 41 317 45 507}	{B28 B30 B31 B8 B5 B28 B26 B29 B24 B24 B20}	{B28}	{B30 B31 B8 B5 B26 B29 B24 B24 B20}	1
57	B28	{495 162 89 330 840 9 41 317 45 507}	{B28 B30 B31 B8 B5 B28 B26 B29 B24 B24 B20}	{A26}	{B30 B31 B8 B5 B26 B29 B24 B24 B20}	1
58	A26	{549 495 162 89 31 330 840 9 41 317 45 507}	{B30 B30 B31 B8 B29 B5 B26 B29 B24 B24 B20}	{B30}	{B31 B8 B29 B5 B26 B29 B24 B24 B20}	1
59	B30	{376 162 89 31 330 41 45 507 122 262 840 9 317}	{A27 A33 A29 A28 A7}	{A27}	{A33 A29 A28 A7}	1
60	A27	{162 89 31 330 41 45 507 122 262 840 9 317}	{A33 A29 A28 A7}	{A33}	{A29 A28 A7}	1
61	A33	{162 89 31 330 41 45 122 262 840 9 317}	{A29 A28 A7}	{A29}	{A28 A7}	1
62	A29	{162 89 31 330 41 45 262 840 9 317}	{A29 A28 A7}	{A28}	{A28 A7}	1
63	A28	{162 89 31 330 41 45 840 9 317}	{B31 B8 B29 B5 B24 B20 B26 B29 B24}	{B31}	{B8 B29 B5 B24 B20 B26 B29 B24}	1
64	B31	{89 31 330 41 45 840 9 317}	{B31 B8 B29 B5 B24 B20 B26 B29 B24}	{A32}	{B8 B29 B5 B24 B20 B26 B29 B24}	1
65	A32	{8 8 8 8 8 8 8 89 31 330 41 45 840 9 317}	{B8 B29 B5 B24 B20 B26 B29 B24}	{B8}	{B29 B5 B24 B20 B26 B29 B24}	1

66	B8	{8 8 8 8 8 8 8 31 330 41 45 840 9 98 317}	{A37 A39 A40 A46 A47 A49 A50 A37 A39 A40 A46 A47 A49 A50 A37 A39 A40 A46 A47 A49 A50 A37 A39 A40 A46 A47 A49 A50 A37 A39 A40 A46 A47 A49 A50 A37 A39 A40 A46 A47 A49 A50 A8}	{A37}	{A39 A40 A46 A47 A49 A50 A39 A40 A46 A47 A49 A50 A39 A40 A46 A47 A49 A50 A39 A40 A46 A47 A49 A50 A39 A40 A46 A47 A49 A50 A39 A40 A46 A47 A49 A50 A8}	1
67	A37	{8 8 8 8 8 8 8 31 330 41 45 840 9 98 317}	{A37 A39 A40 A46 A47 A49 A50 A37 A39 A40 A46 A47 A49 A50 A37 A39 A40 A46 A47 A49 A50 A37 A39 A40 A46 A47 A49 A50 A37 A39 A40 A46 A47 A49 A50 A37 A39 A40 A46 A47 A49 A50 A8}	{A39}	{A39 A40 A46 A47 A49 A50 A39 A40 A46 A47 A49 A50 A39 A40 A46 A47 A49 A50 A39 A40 A46 A47 A49 A50 A39 A40 A46 A47 A49 A50 A39 A40 A46 A47 A49 A50 A8}	1
68	A39	{8 8 8 8 8 8 31 330 41 45 840 9 98 317}	{B29 B5 B24 B20 B26 B29 B24}	{B29}	{B5 B24 B20 B26 B24}	1
69	B29	{8 8 8 8 8 330 41 45 840 98 317}	{A40 A46 A47 A49 A50 A40 A46 A47 A49 A50 A40 A46 A47 A49 A50 A40 A46 A47 A49 A50 A8}	{A40}	{A46 A47 A49 A50 A46 A47 A49 A50 A46 A47 A49 A50 A46 A47 A49 A50 A46 A47 A49 A50 A8}	1

70	A40	{8 8 8 8 330 41 45 840 98 317}	{A40 A46 A47 A49 A50 A40 A46 A47 A49 A50 A40 A46 A47 A49 A50 A40 A46 A47 A49 A50 A40 A46 A47 A49 A50 A8}	{A46}	{A46 A47 A49 A50 A46 A47 A49 A50 A46 A47 A49 A50 A46 A47 A49 A50 A46 A47 A49 A50 A8}	1
71	A46	{8 8 8 330 41 45 840 98 317}	{B5 B24 B20 B26 B24}	{B5}	{B24 B20 B26 B24}	1
72	B5	{262 8 8 8 41 45 840 98 317}	{A7 A47 A49 A50 A47 A49 A50 A47 A49 A50 A8}	{A7}	{A47 A49 A50 A47 A49 A50 A47 A49 A50 A8}	1
73	A7	{68 8 8 8 41 45 840 98 317}	{B6 B24 B20 B26 B24}	{B6}	{B24 B20 B26 B24}	1
74	B6	{8 8 8 41 45 120 840 98 317}	{A47 A49 A50 A47 A49 A50 A47 A49 A50 A8 A8}	{A47}	{A49 A50 A49 A50 A49 A50 A8 A8}	1
75	A47	{8 8 41 45 120 840 98 317}	{A47 A49 A50 A47 A49 A50 A47 A49 A50 A8 A8}	{A48}	{A49 A50 A49 A50 A49 A50 A8 A8}	1
76	A48	{53 8 8 41 45 120 840 98 317}	{B9 B24 B20 B26 B24}	{B9}	{B24 B20 B26 B24}	1
77	B9	{8 8 41 45 120 840 98 317}	{B9 B24 B20 B26 B24}	{A49}	{B24 B20 B26 B24}	1
78	A49	{8 41 45 120 840 98 317}	{B24 B20 B26 B24}	{B24}	{B20 B26}	1
79	B24	{8 45 120 840 98}	{A50 A8 A8}	{A50}	{A8 A8}	1
80	A50	{45 120 840 98}	{A50 A8 A8}	{A8}	{A8 A8}	1
81	A8	{152 45 840}	{B7 B20 B26}	{B7}	{B20 B26}	1
82	B7	{45 840}	{B20 B26}	{B20}	{B26}	1
83	B20	{840}	{B26}	{B26}	{}	0
84	B26	{}	{B26}	{B34}	{}	0
85	B34	{}	{B26}	{B35}	{}	0

BB. Output of Fragment Assembly of pB85766 when difference between numbers of two big bags is low

Node No	CurrBag	S	Options	SelBag	Choices	Cflag
1	A1	{2}	{B1}	{B1}	{}	0
2	B1	{9}	{A2}	{A2}	{}	0
3	A2	{164}	{B2}	{B2}	{}	0
4	B2	{423 105 48}	{A4 A5 A3}	{A4}	{A5 A3}	1
5	A4	{105 48}	{A5 A3}	{A5}	{A3}	1
6	A5	{162 48}	{A5 A3}	{A10}	{A3}	1
7	A10	{50 48 162 10}	{B6 B3 B5}	{B6}	{B3 B5}	1
8	B6	{48 77 162 10}	{B6 B3 B5}	{A11}	{B3 B5}	1
9	A11	{190 114 58 48 162 43 10 44}	{B10 B8 B7 B3 B9 B5 B11}	{B10}	{B8 B7 B3 B9 B5 B11}	1
10	B10	{114 58 48 162 43 10 44}	{B8 B7 B3 B9 B5 B11}	{B8}	{B7 B3 B9 B5 B11}	1
11	B8	{58 48 162 43 10 44}	{B7 B3 B9 B5 B11}	{B7}	{B3 B9 B5 B11}	1
12	B7	{48 162 43 10 44}	{B3 B9 B5 B11}	{B3}	{B9 B5 B11}	1
13	B3	{276 48 44 213 184 43 10}	{A6 A3 A7 A8}	{A6}	{A3 A7 A8}	1
14	A6	{48 44 213 184 43 10}	{A6 A3 A7 A8}	{A12}	{A3 A7 A8}	1
15	A12	{78 48 44 6 213 108 184 43 10}	{B13 B11 B11 B12 B24 B9 B5}	{B13}	{B11 B11 B12 B24 B9 B5}	1
16	B13	{17 48 44 6 213 108 184 43 10}	{B11 B11 B12 B24 B9 B5}	{B11}	{B12 B24 B9 B5}	1
17	B11	{17 48 213 108 184 43 10}	{A13 A3 A7 A19 A8}	{A13}	{A3 A7 A19 A8}	1
18	A13	{48 65 100 213 108 184 421 43 10}	{B16 B14 B12 B24 B15 B9 B5}	{B16}	{B14 B12 B24 B15 B9 B5}	1
19	B16	{262 48 100 213 108 649 184 421 43 10}	{B16 B14 B12 B24 B15 B9 B5}	{A14}	{B14 B12 B24 B15 B9 B5}	1
20	A14	{48 100 213 108 649 184 421 43 10}	{B14 B12 B24 B15 B9 B5}	{B14}	{B12 B24 B15 B9 B5}	1
21	B14	{48 213 108 649 184 421 43 10}	{B14 B12 B24 B15 B9 B5}	{A15}	{B12 B24 B15 B9 B5}	1
22	A15	{48 521 213 108 184 421 43 10 171}	{B18 B12 B24 B15 B9 B5 B17}	{B18}	{B12 B24 B15 B9 B5 B17}	1

23	B18	{48 320 213 108 184 421 312 43 10 171}	{B18 B12 B24 B15 B9 B5 B17}	{A16}	{B12 B24 B15 B9 B5 B17}	1
24	A16	{48 320 213 108 184 421 43 10 171}	{B12 B24 B15 B9 B5 B17}	{B12}	{B24 B15 B9 B5 B17}	1
25	B12	{48 320 213 184 421 43 10 171}	{B12 B24 B15 B9 B5 B17}	{A17}	{B24 B15 B9 B5 B17}	1
26	A17	{466 184 421 319 103 10 48 213 41 43 171}	{B20 B15 B22 B21 B35 B5 B19 B9 B17}	{B20}	{B15 B22 B21 B35 B5 B19 B9 B17}	1
27	B20	{184 421 319 103 10 48 213 41 43 171}	{B15 B22 B21 B35 B5 B19 B9 B17}	{B15}	{B22 B21 B35 B5 B19 B9 B17}	1
28	B15	{184 319 103 10 48 213 41 43 171}	{B22 B21 B35 B5 B19 B9 B17}	{B22}	{B21 B35 B5 B19 B9 B17}	1
29	B22	{184 103 10 48 213 41 117 43 171}	{B21 B35 B5 B19 B9 B17}	{B21}	{B35 B5 B19 B9 B17}	1
30	B21	{184 10 48 213 41 117 43 171}	{B21 B35 B5 B19 B9 B17}	{A18}	{B35 B5 B19 B9 B17}	1
31	A18	{660 184 10 48 235 213 41 43 171}	{B24 B5 B23 B19 B9 B17}	{B24}	{B5 B23 B19 B9 B17}	1
32	B24	{108 305 184 224 10 48 235 213 41 86 43 171}	{A19 A22 A8 A21 A3 A7 A20 A8}	{A19}	{A22 A8 A21 A3 A7 A20 A8}	1
33	A19	{305 184 224 10 48 235 213 41 86 43 171}	{A22 A8 A21 A3 A7 A20 A8}	{A22}	{A8 A21 A3 A7 A20 A8}	1
34	A22	{526 16 55 218 184 224 10 48 96 235 213 41 86 43 171}	{B28 B25 B38 B29 B29 B27 B5 B26 B23 B19 B4 B9 B17}	{B28}	{B25 B38 B29 B29 B27 B5 B26 B23 B19 B4 B9 B17}	1
35	B28	{16 55 218 184 224 10 48 96 235 213 41 86 43 171}	{B25 B38 B29 B29 B27 B5 B26 B23 B19 B4 B9 B17}	{B25}	{B38 B29 B29 B27 B5 B26 B23 B19 B4 B9 B17}	1
36	B25	{55 218 184 224 10 48 96 235 213 41 86 43 171}	{B29 B29 B27 B5 B26 B23 B19 B4 B9 B17}	{B29}	{B27 B5 B26 B23 B19 B4 B9 B17}	1
37	B29	{55 218 184 224 10 48 96 235 213 41 86 43 171}	{B27 B5 B26 B23 B19 B4 B9 B17}	{B27}	{B5 B26 B23 B19 B4 B9 B17}	1
38	B27	{55 184 224 10 48 96 235 213 41 86 43 171}	{B27 B5 B26 B23 B19 B4 B9 B17}	{A20}	{B5 B26 B23 B19 B4 B9 B17}	1
39	A20	{55 184 224 10 48 96 235 213 41 43 171}	{B5 B26 B23 B19 B9 B17}	{B5}	{B26 B23 B19 B9 B17}	1
40	B5	{55 118 184 224 48 96 235 213 41 56 43 171}	{A23 A9 A8 A21 A3 A7 A8}	{A23}	{A9 A8 A21 A3 A7 A8}	1
41	A23	{118 184 224 48 107 174 96 235 213 41 56 43 171}	{B30 B31 B26 B23 B19 B9 B17}	{B30}	{B31 B26 B23 B19 B9 B17}	1

42	B30	{118 184 224 48 174 96 235 213 41 56 43 171}	{B30 B31 B26 B23 B19 B9 B17}	{A21}	{B31 B26 B23 B19 B9 B17}	1
43	A21	{118 184 48 174 96 235 213 41 56 43 171}	{B31 B26 B23 B19 B9 B17}	{B31}	{B26 B23 B19 B9 B17}	1
44	B31	{118 184 48 96 235 36 213 41 209 56 43 171}	{A9 A8 A3 A24 A7 A25 A8}	{A9}	{A8 A3 A24 A7 A25 A8}	1
45	A9	{184 48 96 235 36 213 41 209 56 43 171}	{A8 A3 A24 A7 A25 A8}	{A8}	{A3 A24 A7 A25}	1
46	A8	{86 48 96 235 36 213 41 209 43 171}	{B4 B26 B23 B19 B9 B17}	{B4}	{B26 B23 B19 B9 B17}	1
47	B4	{48 96 235 36 213 41 209 43 171}	{B26 B23 B19 B9 B17}	{B26}	{B23 B19 B9 B17}	1
48	B26	{48 235 36 213 41 209 43 171}	{B26 B23 B19 B9 B17}	{A24}	{B23 B19 B9 B17}	1
49	A24	{48 235 213 41 209 43 171}	{B23 B19 B9 B17}	{B23}	{B19 B9 B17}	1
50	B23	{48 213 41 209 43 171}	{B23 B19 B9 B17}	{A25}	{B19 B9 B17}	1
51	A25	{48 213 41 43 196 171}	{B19 B9 B32 B17}	{B19}	{B9 B32 B17}	1
52	B19	{48 213 43 196 171}	{B9 B32 B17}	{B9}	{B32 B17}	1
53	B9	{48 213 196 171}	{B9 B32 B17}	{A26}	{B32 B17}	1
54	A26	{48 261 213 196 171}	{B32 B32 B17}	{B32}	{B17}	1
55	B32	{21 322 288 150 48 213 510 171}	{A31 A45 A28 A29 A30 A3 A7 A27}	{A31}	{A45 A28 A29 A30 A3 A7 A27}	1
56	A31	{322 288 150 48 213 60 510 171}	{A28 A29 A30 A3 A7 A27}	{A28}	{A29 A30 A3 A7 A27}	1
57	A28	{288 150 48 213 60 510 171}	{A29 A30 A3 A7 A27}	{A29}	{A30 A3 A7 A27}	1
58	A29	{150 48 213 60 510 171}	{A30 A3 A7 A27}	{A30}	{A3 A7 A27}	1
59	A30	{48 213 60 510 171}	{A3 A7 A27}	{A3}	{A7 A27}	1
60	A3	{213 60 510 171}	{A3 A7 A27}	{A27}	{A7 A27}	1
61	A27	{213 60 171}	{B33 B17}	{B33}	{B17}	1
62	B33	{3 213 81 171}	{A33 A7 A32}	{A33}	{A7 A32}	1
63	A33	{103 28 294 213 81 171}	{B35 B36 B34 B17}	{B35}	{B36 B34 B17}	1
64	B35	{28 294 213 81 171}	{B36 B34 B17}	{B36}	{B34 B17}	1
65	B36	{116 57 294 213 81 171}	{A35 A34 A7 A32}	{A35}	{A34 A7 A32}	1

66	A35	{33 57 294 81 213 171}	{A35 A34 A7 A32}	{A32}	{A34 A7 A32}	1
67	A32	{33 57 294 213 171}	{B37 B34 B17}	{B37}	{B34 B17}	1
68	B37	{57 294 507 254 213 171}	{A34 A36 A37 A7}	{A34}	{A36 A37 A7}	1
69	A34	{294 507 254 213 171}	{A36 A37 A7}	{A36}	{A37 A7}	1
70	A36	{294 254 213 171}	{A36 A37 A7}	{A37}	{A37 A7}	1
71	A37	{294 75 213 171}	{B34 B38 B17}	{B34}	{B38 B17}	1
72	B34	{75 213 171}	{B38 B17}	{B38}	{B17}	1
73	B38	{16 63 779 189 282 213 160 171}	{A43 A42 A38 A39 A41 A7 A40}	{A43}	{A42 A38 A39 A41 A7 A40}	1
74	A43	{63 779 189 282 213 160 4 171}	{A42 A38 A39 A41 A7 A40}	{A42}	{A38 A39 A41 A7 A40}	1
75	A42	{779 189 282 213 160 4 171}	{A38 A39 A41 A7 A40}	{A38}	{A39 A41 A7 A40}	1
76	A38	{189 282 213 160 4 171}	{A39 A41 A7 A40}	{A39}	{A41 A7 A40}	1
77	A39	{282 213 160 4 171}	{A41 A7 A40}	{A41}	{A7 A40}	1
78	A41	{213 160 4 171}	{A41 A7 A40}	{A40}	{A7 A40}	1
79	A40	{213 4 171}	{B39 B17}	{B39}	{B17}	1
80	B39	{21 95 213 171}	{A45 A44 A7}	{A45}	{A44 A7}	1
81	A45	{95 289 213 171}	{A45 A44 A7}	{A44}	{A44 A7}	1
82	A44	{289 213 171}	{B40 B17}	{B40}	{B17}	1
83	B40	{213 171}	{B40 B17}	{A7}	{B17}	1
84	A7	{171}	{B17}	{B17}	{}	0